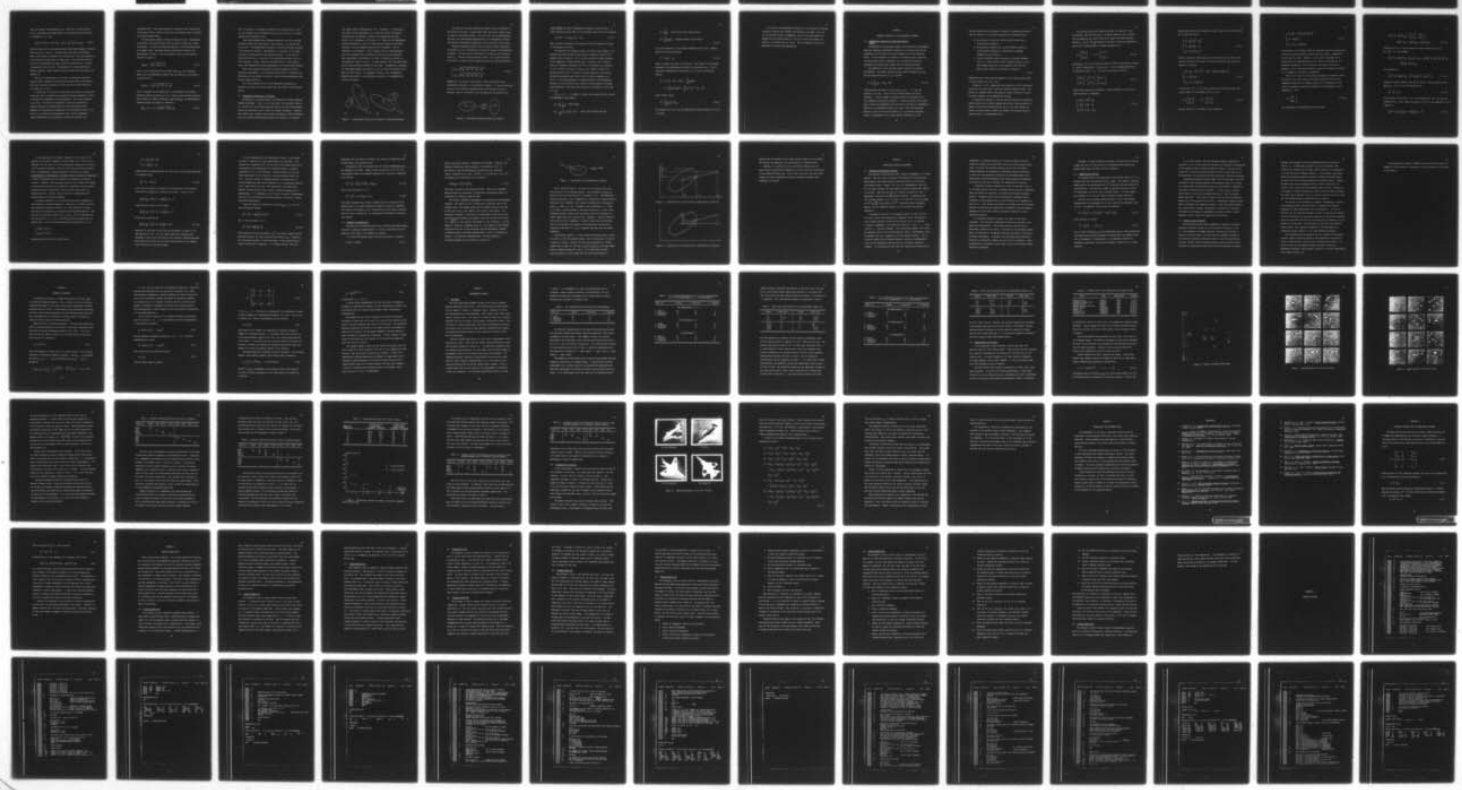
AD-A058 466 TENNESSEE UNIV KNOXVILLE DEPT OF ELECTRICAL ENGINEERING F/G 9/2
A RECURSIVE PROCEDURE FOR UPDATING QUADRATIC FORMS AND ITS APPL--ETC(U)
JAN 78 R C GONZALEZ, B M MORET N00014-75-C-0545

UNCLASSIFIED ONR-CR215-288-3F NL

1 OF 3
ADA
058466



REPORT ONR-CR215-288-3F

LEVEL II

OFFICE OF NAVAL RESEARCH

A RECURSIVE PROCEDURE FOR UPDATING QUADRATIC FORMS AND ITS APPLICATION TO FEATURE SELECTION

R.C. GONZALEZ and B.M.E. MORET

ELECTRICAL ENGINEERING DEPARTMENT
UNIVERSITY OF TENNESSEE
KNOXVILLE, TN 37916

CONTRACT N00014-75-C-0545
ONR TASK 215-228

30 JANUARY 1978

210p.

REPORT FOR THE PERIOD 1 OCT 76 - 1 OCT 77

Rept. for 1 Oct 76 - 1 Oct 77,

78 06 15 054

PREPARED FOR THE

OFFICE OF NAVAL RESEARCH • 800 N. QUINCY ST. • ARLINGTON • VA • 22217

404 468

ABSTRACT

A survey of the problem of the selection of measurements is presented. The Mahalanobis distance is introduced as a selection criterion, and approached from an algebraic point of view. A new method for updating the inverse of a positive definite matrix is derived, and applied to the selection problem. Two interactive procedures are developed and used in experiments with the classic problem of Fisher, with high-resolution radar images, and with aircraft photographs. Moments are used as features in the last two experiments.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATION AND TERMS

$\underline{x}$ (lower case underlined letter)	:	column vector
$\underline{x}^T$	:	transpose of $\underline{x}$
$\|\|\underline{x}\|\|_2$ (= $\underline{x}^T \cdot \underline{x}$)	:	natural 2-norm of $\underline{x}$
$\underline{m}_i$	:	mean vector of the i-th class
$\underline{A}$ (upper case underlined letter)	:	matrix
$\underline{A}^T$	:	transpose of $\underline{A}$
$\|\underline{A}\|$	:	determinant of $\underline{A}$
$\underline{A}^{-1}$	:	inverse of $\underline{A}$
$\underline{C}_i$	:	covariance matrix of the i-th class
$\underline{F}$	:	complex Fourier matrix
$\underline{F}^H$	:	Hermitean transpose of $\underline{F}$
ω_i	:	i-th class (a set of patterns)
$p(\underline{x})$	:	probability density function of $\underline{x}$
$p(\underline{x}/\omega_i)$	:	conditional probability density function of $\underline{x}$ given ω_i
$r_j(\underline{x})$	:	average loss incurred by classifying $\underline{x}$ in class ω_j

Unless otherwise specified, all numbers, vectors, and matrices are real, and all matrices are square.

The terms "measurement" and "component" are used interchangeably; likewise, the terms "pattern" and "(measurement) vector" are substituted indifferently for one another.

As used in the text, a principal submatrix of a matrix $\underline{A}$ is a matrix of dimension inferior or equal to the dimension of $\underline{A}$, and the diagonal elements of which are diagonal elements of $\underline{A}$.

CHAPTER 1

INTRODUCTION

The recognition of patterns is a basic act of human life, indeed, of animal life. The duplication of this activity in a machine has been the object of numerous studies, and the number of publications dealing with this problem has been steadily increasing over the past decade.

Three stages can be distinguished in a typical pattern recognition process: feature extraction, measurement selection, and measurement classification. This study attempts to provide a new solution to the second problem, the selection of measurements. Since no stage can be examined independently of the others, consideration is given to the other two problems as well, but only in the measure where they influence decision processes in the selection problem. Various authors have extensively surveyed the field of pattern recognition ([14], [19], [23]). A detailed review of contributions to the subject can be found in an article by Das Gupta [9].

CHAPTER 2

SELECTION OF MEASUREMENTS

2.1. Desirability of a Selection

The question might arise as to why a selection of measurements is desirable. As answers to this question are of obvious importance in the design of a selection scheme, some are presented below.

First, not all the measurements are necessarily useful to classification. Indeed, if the distribution of a measurement is the same for all pattern classes, that measurement is useless for the purpose of discrimination. Though this can be avoided by a careful design of the feature extraction scheme, a priori information about the data to be classified may be insufficient, or the features may be too difficult to interpret (as in the case discussed in Chapter 5).

Secondly, an excess of measurements can cause difficulties during a training process. As only a limited set of patterns—the training set—is available during training, the number of measurements can reach or exceed the number of training samples. This results in a classifier being sensitive to the separation of sample patterns instead of to the distributions the latter should represent. In the particular case of a linear decision function, this corresponds to the fact that, if the patterns are well-distributed (i.e., in general position), there exists a linear dichotomy separating any two given distinct subsets of the training set [21, pp. 58-60]. As an enlargement of the training set is not always possible, it may be necessary to reduce the number of measurements.

A third answer is economy. Processing a large number of measurements is costly in terms of computation time, storage memory, and communication channels. Likewise, performing the measurements might require expensive sensors and exert a perturbing, if not disrupting, influence on the process to be measured. Thus it is desirable that the number of measurements be limited to the minimum necessary to insure proper classification. In fact, one is often forced to trade low error rates for reduced number of measurements.

2.2. Selection Criteria and Methods

In order to carry out a selection procedure, some criterion of choice is needed. In general, given a set of N measurements, a subset of N_1 ($N_1 < N$) measurements will be chosen such that no other subset yields better results, according to some definition of a "good result." This problem is often constrained by the condition that the chosen subset does not contain more than a given limit number of measurements.

The selection procedure depends on the definition of a "good subset" which, in turn, often depends on the type of classifier to be ultimately used. Consequently, there exists a variety of criteria, each resulting in a different selection procedure. Some of these are examined below.

The optimal strategy can be defined as that incurring the minimum loss. This is the basis of the Bayes strategy, widely used in pattern classification [21, Chapter 4]. When such a classifier is to be used, the selection criterion is naturally derived from it. Therefore, a subset of measurements will be said to be better than, and will consequently be preferred to, another subset if the average loss

associated with its choice is less than that associated with the choice of the other subset. The average loss incurred when classifying a pattern can be expressed as follows. Let M denote the number of classes, ω_i be the set of all patterns pertaining to class i, and $\underline{x}$ be an unknown measurement vector; finally, let L_{ij} denote the loss incurred when classifying a pattern of class i into class j. Then the average loss associated with the classification of a pattern into class j is given by:

$$r_j(\underline{x}) = \sum_{i=1}^{M} L_{ij}\, p(\omega_i/\underline{x}) \qquad (2.2\text{-}1)$$

and the classification strategy is to assign $\underline{x}$ to class j if $r_j(\underline{x}) < r_i(\underline{x})$, for all i different from j. The L_{ij} coefficients can usually be taken as:

$$L_{ij} = 1 - \delta_{ij} \qquad (2.2\text{-}2)$$

where δ_{ij} is the Kronecker function; (2.2-1) then reduces to:

$$r_j(\underline{x}) = \frac{1}{p(\underline{x})} \sum_{i=1}^{M} (1 - \delta_{ij}) p(\underline{x}/\omega_i) p(\omega_i)$$

$$= 1 - \frac{1}{p(\underline{x})}\, p(\underline{x}/\omega_j) p(\omega_j) \qquad (2.2\text{-}3)$$

where use has been made of Bayes' formula. When used for classification,

$r_j(\underline{x})$ can be simplified by dropping the terms independent from the class index. Thus (2.2-3) finally becomes:

$$r_j'(\underline{x}) = -p(\underline{x}/\omega_j) \cdot p(\omega_j) \qquad (2.2\text{-}4)$$

It is noted, however, that the assumption of (2.2-2) cannot always be made. As an example, consider a medical application where the classifier attempts to distinguish healthy persons from unhealthy ones. Clearly, the loss associated with the classification of an ill subject as healthy is much higher than that associated with the reverse case. In such cases, equation (2.2-1) must be retained in spite of its higher complexity.

The problem encountered in implementing a Bayes classifier is the estimation of the probability density functions, $p(\omega_i)$ and $p(\underline{x}/\omega_i)$. This problem becomes even worse in a selection process as the measurement vector, $\underline{x}$, is then of variable dimensionality. Indeed, the problem is, in most cases, intractable. However, solutions exist in the case of some special distributions, in particular normal and Poisson distributions [1]. It will be recalled that a normal, or Gaussian, process is entirely characterized by its mean and covariance matrix (i.e., by statistical moments of the first and second order). Let $\underline{m}$ be the mean vector and $\underline{C}$ the covariance matrix; then the probability density function of a normal distribution is given by:

$$p(\underline{x}) = 2(\pi)^{-n/2} \, |\underline{C}|^{-1/2} \exp \left\{ -\frac{1}{2} (\underline{x} - \underline{m})^T \, \underline{C}^{-1} \, (\underline{x} - \underline{m}) \right\} \qquad (2.2\text{-}5)$$

where $|\underline{C}|$ denotes the determinant of $\underline{C}$. When this is substituted in (2.2-4), and after some manipulations, the following decision function is obtained [21, p. 121]:

$$d_j(\underline{x}) = \ln p(\omega_j) - \frac{1}{2} \ln |\underline{C}_j| - \frac{1}{2} (\underline{x} - \underline{m}_j)^T \underline{C}_j^{-1} (\underline{x} - \underline{m}_j) \qquad (2.2\text{-}6)$$

where the mean vector and covariance matrix have been indexed to indicate their relation to class j. Both the mean vector and the covariance matrix can be easily estimated from a training set; the class probability can be derived by observation of sample data. Thus the Bayes decision rule can be implemented for normal pattern distribution with a minimum of complexity (see [2], [9]). The probability of misclassification, however, remains a highly complex function, unless more restrictions are imposed [7].

Thus, most investigations in the field of measurement selection based on Bayes' probability of error have assumed normal distributions with equal covariance matrices—in which case the decision functions are linear [21, p. 121].

A second type of criterion is based upon the results of information theory and tends to minimize the correlation between components in order to minimize the redundancy of the measurements. This can be achieved in two ways. The first is not properly a selection procedure but rather a reduction of dimensionality and uses the Karhunen-Loève expansion to decorrelate the components of the measurement vector. The result is a rotation in the measurement space; the new components, linear combinations of the originals, can then be selected on an

individual basis. The second approach is sequential and, starting with an arbitrary choice, selects at each step the component least correlated with those already chosen.

The third major category includes distance criteria. Measurements are selected which maximize some type of distance between the classes of patterns. In order of increasing complexity, the following measures are commonly used: Euclidean distance, Mahalanobis distance, and Bhattacharyya distance. Given two patterns, $\underline{x}$ and $\underline{y}$, the Euclidean distance is given by:

$$d_E(\underline{x},\underline{y}) = \sqrt{(\underline{x} - \underline{y})^T(\underline{x} - \underline{y})} \tag{2.2-7}$$

Given a class characterized by its mean vector, $\underline{m}_i$, and covariance matrix, $\underline{C}_i$, the Mahalanobis distance from the pattern $\underline{x}$ to the class i is given [17] by:

$$d_M(\underline{x},\omega_i) = \sqrt{(\underline{x} - \underline{m}_i)^T \underline{C}_i^{-1} (\underline{x} - \underline{m}_i)} \tag{2.2-8}$$

This is recognized as the square root of the exponent of the normal probability density function of (2.2-5). Finally, given two classes given by their probability density functions, $p_1(\underline{x})$ and $p_2(\underline{x})$, the Bhattacharyya distance between the classes is defined as:

$$d_B(p_1, p_2) = -\ell n \int \sqrt{p_1(\underline{x}) \cdot p_2(\underline{x})}\ d\underline{x} \tag{2.2-9}$$

This is a measure of divergence between the two distributions, p_1 and p_2; the integral can be interpreted as giving the cosine of the "angle" between the distributions [19].

These three measures have different properties and do not measure distances between the same objects. Each, however, is a special case of the next. The Bhattacharyya distance is the most efficient criterion of the three for selection. Unfortunately, it requires, as the Bayes' criterion, the estimation of the density functions. In the case of normal distributions, it reduces to the Mahalanobis distance, which is more tractable. Finally, when the covariance matrix is the identity matrix, the Mahalanobis distance degenerates into the Euclidean distance.

Other, more specialized criteria have been proposed to fulfill particular requirements. All of these criteria result, in practice, in selections that are better—in terms of classification performance—than trivial random selections.

This study proposes the use of the Mahalanobis distance as a selection criterion for several reasons that are noted in the following section.

2.3. Mahalanobis Distance as a Criterion

Considered as a distance measure, the Mahalanobis distance presents numerous advantages. First, it is a true metric in real space; that is, it is positive (or zero, which correctly implies that the two considered vectors are identical), and it obeys the triangle inequality. Moreover, this measure stays invariant under general nonsingular linear transformations [6], whereas the Euclidean distance, for instance, is invariant

only under unitary transformations (i.e., rotations). In particular, this makes scaling unnecessary, as it would not affect the measure. (Scaling might be necessary, however, for computational reasons.) Finally, as a consequence of the above, it is seen that the Mahalanobis distance measure has a "built-in" scaling which weighs the Euclidean distance by the set variance along the axis of measure. This is important in two respects. First, measures taken in disproportioned units are considered only as relative variations and thus rescaled. More significantly, the spread of the sets is taken into account, as illustrated in Figure 1a and 1b. In these figures, the three mean points are at fixed Euclidean distances of each other. Consequently, according to a Euclidean measure, the pattern P would be said to be closer to the mean M_1 in both cases. It is apparent, however, that a Mahalanobis measure would result in two different conclusions, in accordance with intuitive reasoning.

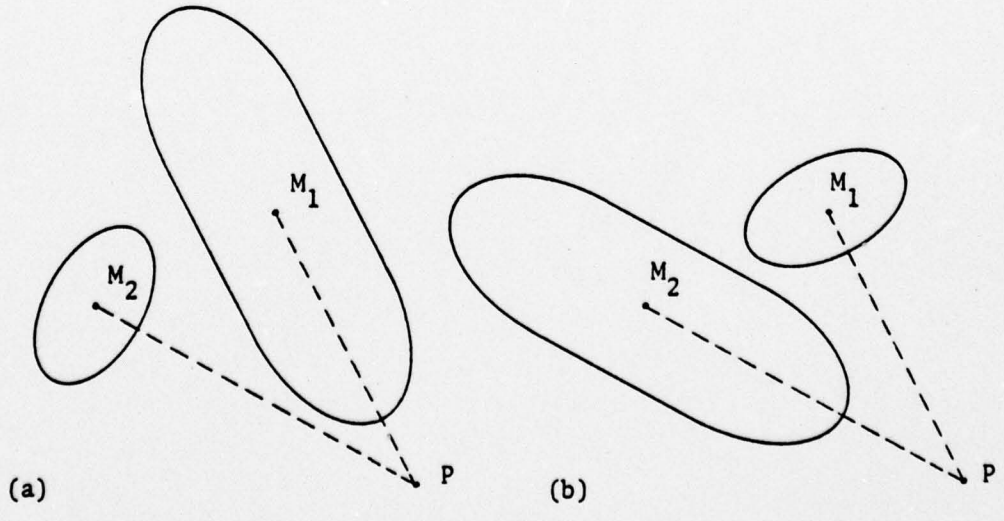


Figure 1. Illustration of the role of spread in distance measurement.

Thus the use of second statistical moments allows a refinement of the notion of distance. Though higher order statistical moments could be used (for non-Gaussian data), as is the case with the Bhattacharyya distance, the second-degree approximation of the Mahalanobis distance constitutes a satisfactory description of the situation in most cases.

Selection procedures usually involve several classes of patterns. The problem is how to characterize interclass distances with Mahalanobis distances. Let each class be described by its mean $\underline{m}_i$ and covariance matrix $\underline{C}_i$. Then the distance between classes i and j can be measured in two ways, in terms of their respective means and covariance matrices.

$$\begin{cases} d_{ij} = \sqrt{(\underline{m}_i - \underline{m}_j)^T \, \underline{C}_i^{-1} \, (\underline{m}_i - \underline{m}_j)} \\ d_{ji} = \sqrt{(\underline{m}_i - \underline{m}_j)^T \, \underline{C}_j^{-1} \, (\underline{m}_i - \underline{m}_j)} \end{cases} \tag{2.3-1}$$

Depending on the spread of the classes, these two measures can be considerably different, as illustrated in Figure 2. A better characterization of the interclass distance would be the average of the two measures, that is, the double of the distance from the median of the

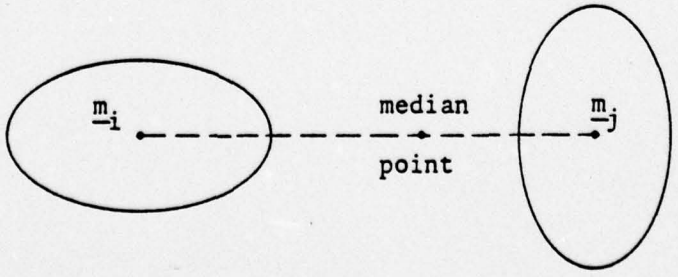


Figure 2. Mahalanobis distance between two classes.

segment $\overline{\underline{m}_i \, \underline{m}_j}$ (in terms of Mahalanobis distance) to either class. A rapid calculation shows this to be the harmonic mean of the two distances:

$$d_{ij} \text{ aver.} = 2\, d_{ji} d_{ij} / (d_{ji} + d_{ij}) \qquad (2.3\text{-}2)$$

Thus, as would be expected, the average of the two distances is closer to the smaller of the two.

For more than two classes, however, the case is more complex. The average interclass distance should possess a property similar to the harmonic mean of equation (2.3-2) so that excessively large distances do not completely override smaller ones. There are, for N classes, $N \cdot (N-1)/2$ interclass distances (or $N \cdot (N-1)$ if measured as in equation (2.3-1)). There is no point in using (2.3-2) anymore; rather, two possibilities are left. The first is almost simplistic and involves taking the arithmetic mean of the $N \cdot (N-1)$ distances computed as in (2.3-1). The second approach is a generalization of the Mahalanobis distance to the multiclass case, yielding a unique measure of average class separation, based upon scatter matrices introduced by Wilks (see [1], [6]).

Let ω_i, i=1, ..., M, denote M classes, each comprised of N_i vectors of measurements; then define:

$$\underline{m}_i = \frac{1}{N_i} \sum_{\underline{x} \in \omega_i} \underline{x} \quad \text{mean vectors}$$

$$\underline{W}_i = \sum_{\underline{x} \in \omega_i} (\underline{x} - \underline{m}_i)(\underline{x} - \underline{m}_i)^T \quad \text{within-class scatter matrices}$$

$$\underline{W} = \sum_{i=1}^{M} \underline{W}_i \quad \text{total within-class scatter matrix}$$

$$\underline{B} = \sum_{i=1}^{M} N_i \underline{m}_i \underline{m}_i^T \quad \text{between-classes scatter matrix} \tag{2.3-3}$$

$\underline{W}$ is also referred to as the pooled covariance matrix [10]. Finally, define the following measure:

$$d^2 = \text{tr}(\underline{W}^{-1} \cdot \underline{B}) \tag{2.3-4}$$

where tr denotes the trace of the matrix. This measure is the natural extension of the Mahalanobis distance to a multiclass case. This is more easily understood by noting that (2.3-4) can be rewritten as follows:

$$d^2 = \text{tr}\ (\underline{W}^{-1} \cdot \underline{B}) = \text{tr}(\underline{W}^{-1} \cdot \sum_{i=1}^{M} N_i \underline{m}_i \underline{m}_i^T)$$

$$= \text{tr}\ (\sum_{i=1}^{M} N_i \underline{W}^{-1} \underline{m}_i \underline{m}_i^T) = \sum_{i=1}^{M} N_i \text{tr}\ (\underline{W}^{-1} \cdot \underline{m}_i \cdot \underline{m}_i^T)$$

which finally yields:

$$d^2 = \sum_{i=1}^{M} N_i \underline{m}_i^T \underline{W}^{-1} \underline{m}_i \tag{2.3-5}$$

The analogy of (2.3-5) with the Mahalanobis distance defined in (2.2-8) is evident.

The selection of measurements according to the criterion of maximum interclass distance then requires the evaluation of either (2.3-1) or (2.3-5) for each subset of components; this in turn necessitates the updating of the inverse matrix $\underline{C}^{-1}$ (or $\underline{W}^{-1}$) as rows and columns are added to, or deleted from, $\underline{C}_i$ (or $\underline{W}$). Thus an algebraic study of the Mahalanobis distance seems appropriate.

CHAPTER 3

ALGEBRAIC APPROACH TO THE MAHALANOBIS DISTANCE

3.1. Updating the Inverse of a Positive-Definite Matrix

As mentioned in the previous chapter, the selection of measurements using the interclass Mahalanobis distance as a criterion involves the computation of the inverse of various submatrices of the covariance matrix. As the inversion of a matrix is a lengthy process, some kind of recursiveness would be desirable; this would have the further advantage of providing some insight into the selection problem.

The properties of the covariance matrix simplify the problem considerably. The sample covariance matrix may be computed by using the following definition [21, pp. 137-139]:

$$\underline{C} = \frac{1}{N} \sum_{i=1}^{N} (\underline{x}_i - \underline{m})(\underline{x}_i - \underline{m})^T \qquad (3.1\text{-}1)$$

where $\underline{m}$ denotes the mean of the set and $\underline{x}_i$, i=1, ..., N, are the patterns in the set. (This is the so-called maximum likelihood estimator.) Thus the sample covariance matrix is guaranteed to be real symmetric and positive semidefinite (since any matrix of the type $\underline{A} \cdot \underline{A}^T$ is positive semidefinite and since the sum of positive semidefinite matrices is a semidefinite matrix [5], [18]). The true covariance matrix is known to be positive definite [1]; the sample covariance matrix is guaranteed to be so under certain conditions [1], [8].

Positive definite matrices possess a number of interesting properties; those of interest to the present discussion are enumerated below:

-a- The inverse of a positive definite matrix exists and is itself positive definite

-b- Any principal submatrix of a positive definite matrix is itself positive definite; in particular all diagonal elements are real positive

-c- A positive-definite matrix possesses a Cholesky decomposition; i.e., there exists a lower triangular matrix $\underline{L}$, with real elements (positive on the diagonal) such that the product $\underline{L} \cdot \underline{L}^T$ is equal to the matrix in question.

(3.1-2)

Demonstrations of these various properties can be found in most books on linear algebra [5], [18], [22].

Property (3.1-2-b-) guarantees that submatrices of the covariance matrix will also give rise to positive definite quadratic forms. This proves the statement made in the previous chapter (section 2.3) about the metric property of the Mahalanobis distance.

Of particular interest is the fact that the inverse of a positive-definite matrix can be easily updated when either a rank-one matrix is added to the positive definite matrix [4] or a row and a column (or several rows and several columns) are added to the original positive definite matrix, as demonstrated below.

Let $\underline{C}$ and $\underline{C}_k$ be positive definite matrices of dimension n and k respectively, and such that $\underline{C}_k$ is a principal submatrix of $\underline{C}$. Without loss of generality, $\underline{C}_k$ can be taken as the upper left-hand principal submatrix. Let $\underline{C}^{-1}$ and $\underline{C}_k^{-1}$ denote the inverse of $\underline{C}$ and $\underline{C}_k$ respectively, and let $\underline{C}^{-1}$ be partitioned in a manner analogous to $\underline{C}$:

$$\underline{C} = \begin{bmatrix} \underline{C}_k & \underline{X} \\ \underline{X}^T & \underline{Y} \end{bmatrix} \quad , \quad \underline{C}^{-1} = \begin{bmatrix} \underline{U} & \underline{V} \\ \underline{V}^T & \underline{W} \end{bmatrix} \tag{3.1-3}$$

By property (3.12-a-b) $\underline{Y}$ and $\underline{W}$ are positive definite and possess inverses, $\underline{Y}^{-1}$ and $\underline{W}^{-1}$ respectively. $\underline{X}$ and $\underline{V}$ are $k \times (n-k)$ rectangular matrices. By definition, $\underline{C} \cdot \underline{C}^{-1} = \underline{I}_n$. This can be rewritten as follows:

$$\begin{bmatrix} \underline{C}_k & \underline{X} \\ \underline{X}^T & \underline{Y} \end{bmatrix} \cdot \begin{bmatrix} \underline{U} & \underline{V} \\ \underline{V}^T & \underline{W} \end{bmatrix} = \begin{bmatrix} \underline{I}_k & \underline{O} \\ \underline{O} & \underline{I}_{n-k} \end{bmatrix} \tag{3.1-4}$$

which yields three matrix equations in three unknowns, $\underline{U}$, $\underline{V}$, and $\underline{W}$ (a fourth equation is redundant).

$$\begin{cases} \underline{C}_k \cdot \underline{U} + \underline{X} \cdot \underline{V}^T = \underline{I}_k \\ \underline{C}_k \cdot \underline{V} + \underline{X} \cdot \underline{W} = \underline{O} \\ \underline{X}^T \cdot \underline{V} + \underline{Y} \cdot \underline{W} = \underline{I}_{n-k} \end{cases} \tag{3.1-5}$$

Multiplying the first two equations by $\underline{C}_k^{-1}$ on the left, and the third by $\underline{Y}^{-1}$ on the left yields:

$$\begin{cases} \underline{U} + \underline{C}_k^{-1} \cdot \underline{X} \cdot \underline{V}^T = \underline{C}_k^{-1} \\ \underline{V} + \underline{C}_k^{-1} \cdot \underline{X} \cdot \underline{W} = \underline{O} \\ \underline{W} + \underline{Y}^{-1} \cdot \underline{X}^T \cdot \underline{V} = \underline{Y}^{-1} \end{cases} \tag{3.1-6}$$

Finally, using the second equation in the third and in the first, and since $\underline{W}$ is known to exist and to be positive definite, the following solution is obtained:

$$\begin{cases} \underline{U} = (\underline{C}_k - \underline{X} \cdot \underline{Y}^{-1} \underline{X}^T)^{-1} = \underline{C}_k^{-1} + (\underline{C}_k^{-1} \cdot \underline{X}) \cdot \underline{W} \cdot (\underline{C}_k^{-1} \cdot \underline{X})^T \\ \underline{V} = -\underline{C}_k^{-1} \cdot \underline{X} \cdot \underline{W} \\ \underline{W} = (\underline{Y} - \underline{X}^T \cdot \underline{C}_k^{-1} \cdot \underline{X})^{-1} \end{cases} \tag{3.1-7}$$

In particular, if k = n-1, $\underline{X}$ and $\underline{V}$ reduce to the vectors $\underline{x}$ and $\underline{v}$ and $\underline{Y}$ and $\underline{W}$ reduce to the scalars α and β so that:

$$\underline{C} = \begin{bmatrix} \underline{C}_k & \underline{x} \\ \underline{x}^T & \alpha \end{bmatrix} \quad , \quad \underline{C}^{-1} = \begin{bmatrix} \underline{U} & \underline{v} \\ \underline{v}^T & \beta \end{bmatrix} \tag{3.1-8}$$

and the system (3.1-7) reduces to the following:

$$\begin{cases} \underline{U} = \underline{C}_k^{-1} + \beta \cdot (\underline{C}_k^{-1} \cdot \underline{x}) \cdot (\underline{C}_k^{-1} \cdot \underline{x})^T \\ \underline{v} = -\beta \cdot \underline{C}_k^{-1} \cdot \underline{x} \\ \beta = 1/(\alpha - \underline{x}^T \cdot \underline{C}_k^{-1} \cdot \underline{x}) \end{cases} \tag{3.1-9}$$

The system (3.1-9) provides a fast and efficient method for updating the inverse of a positive-definite matrix when the latter is augmented by one row and one column. Moreover, if no a priori knowledge can be assumed about the augmented matrix, the value of β provides all the necessary information: if β is positive, the matrix is positive definite; if β is infinite, the matrix is positive semidefinite singular; if β is negative, the matrix is indefinite.

Both (3.1-7) and (3.1-9) can easily be applied to quadratic forms. Let $\underline{m}$ be an n×1 vector and $\underline{m}_k$ a k×1 vector formed with the k first components (again this is assumed without loss of generality as the value of a quadratic form does not depend on the arrangement of the components). Thus:

$$\underline{m} = \begin{bmatrix} \underline{m}_k \\ \underline{m}_{n-k} \end{bmatrix} \tag{3.1-10}$$

Now the quadratic form $\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m}$ can be rewritten:

$$\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} = \begin{bmatrix} \underline{m}_k^T & \underline{m}_{n-k}^T \end{bmatrix} \cdot \begin{bmatrix} \underline{U} & \underline{V} \\ \underline{V}^T & \underline{W} \end{bmatrix} \cdot \begin{bmatrix} \underline{m}_k \\ \underline{m}_{n-k} \end{bmatrix}$$

$$= \underline{m}_k^T \cdot \underline{U} \cdot \underline{m}_k + 2\, \underline{m}_k^T \cdot \underline{V} \cdot \underline{m}_{n-k} + \underline{m}_{n-k}^T \cdot \underline{W} \cdot \underline{m}_{n-k} \,. \qquad (3.1\text{-}11)$$

Making use of (3.1-7) and writing the Cholesky decomposition of $\underline{W}$: $\underline{W} = \underline{L} \cdot \underline{L}^T$, (3.1-11) becomes:

$$\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} = \underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k + \underline{m}_{n-k}^T \cdot \underline{L} \cdot \underline{L}^T \cdot \underline{m}_{n-k} + \underline{m}_k^T \cdot (\underline{C}_k^{-1} \cdot \underline{X}) \cdot \underline{L} \cdot \underline{L}^T \cdot (\underline{C}_k^{-1} \cdot \underline{X})^T \cdot \underline{m}_k$$

$$- 2 \cdot \underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{X} \cdot \underline{L} \cdot \underline{L}^T \cdot \underline{m}_{n-k}$$

which reduces to:

$$\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} = \underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k + \| \underline{L}^T \cdot (\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{X} - \underline{m}_{n-k}^T)^T \|_2^2 \qquad (3.1\text{-}12)$$

Denoting by d_n^2 the quadratic form $\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m}$ and by d_k^2 the quadratic form $\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k$, (3.1-12) can be reformulated as:

$$d_n^2 = d_k^2 + \Delta_{n-k}^2 \qquad (3.1\text{-}13)$$

The reduction to the case $k = n-1$ is straightforward. Let μ be the last component of $\underline{m}$. Then, using the notation of (3.1-9), the equation (3.1-12) reduces to:

$$\underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} = \underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k + \beta \cdot (\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{x} - \mu)^2 \qquad (3.1\text{-}14)$$

To the knowledge of the author, equations (3.1-12) and (3.1-14) represent a new result, together with the systems (3.1-7) and (3.1-9). Equations (3.1-12) and (3.1-14) are particularly significant in relation with the selection problem: they show that positive definite quadratic forms, the Mahalanobis distance (squared) in particular, increase monotonically as new components are added (thus the process of selecting a subset of components can only decrease the separation between the classes). Moreover, these equations provide a way to compute the change in distance associated with the addition of components without actually computing the new inverse matrix, thus obviously saving time in a selection process. Finally, once a selection is accomplished, the inverse matrix can be rapidly updated.

The matrix equations (3.1-4) have been solved for an updating in the direction of increased dimension. Though this is likely to be the most commonly encountered situation, it is also desirable to derive solutions for an updating in the direction of decreased dimension. Thus $\underline{C}_k^{-1}$ must be expressed in terms of $\underline{U}$, $\underline{V}$, and $\underline{W}$. This is presented only for the case of k = n-1, as it is of more practical interest than the general case. Two of the three equations are rewritten for k = n-1:

$$\begin{cases} \underline{C}_k \cdot \underline{U} + \underline{x} \cdot \underline{v}^T = \underline{I} \\ \underline{C}_k \cdot \underline{v} + \beta \cdot \underline{x} = \underline{0} \end{cases}$$

Multiplying both on the left by $\underline{C}_k^{-1}$ yields:

$$\begin{cases} \underline{U} + \underline{C}_k^{-1} \cdot \underline{x} \cdot \underline{v}^T = \underline{C}_k^{-1} \\ \underline{v} + \beta \cdot \underline{C}_k^{-1} \cdot \underline{x} = \underline{0} \end{cases}$$

Substituting the second equation into the first, the following expression is obtained for $\underline{C}_k^{-1}$:

$$\underline{C}_k^{-1} = \underline{U} - 1/\beta \cdot \underline{v} \cdot \underline{v}^T \qquad (3.1\text{-}15)$$

This result can readily be applied to the evaluation of the quadratic form—where now $\underline{m}_k^T \underline{C}_k^{-1} \underline{m}_k$ is desired—as follows. From (3.1-14):

$$\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k = \underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} - \beta (\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{x} - \mu)^2$$

Using the above results, this becomes:

$$\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k = \underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} - \beta (-\underline{m}_k^T \cdot \frac{1}{\beta} \cdot \underline{v} - \mu)^2$$

which can be rewritten as:

$$\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{m}_k = \underline{m}^T \cdot \underline{C}^{-1} \cdot \underline{m} - \frac{1}{\beta} (\underline{m}_k^T \cdot \underline{v} + \beta\mu)^2 \qquad (3.1\text{-}16)$$

Equations (3.1-15) and (3.1-16) are the counterpart of system (3.1-9) and equation (3.1-14). All the remarks made above concerning the advantages of the latter also hold for the "downward" updating procedure. Thus these results allow for an overall simplification of the computations involved in a selection process.

As the incorporation of all measurements results in the maximum distance, a computation of this upper bound may be desirable. This requires the evaluation of $\underline{C}^{-1}$; as the value of the upper bound should be available at the beginning of the selection process, a stepwise computation of $\underline{C}^{-1}$ is not efficient. Several methods exist for the direct calculation of the inverse of a positive-definite matrix [6], [22]. Once this inverse is obtained, however, the computation of inverse submatrices might not be necessary, as a corresponding submatrix of $\underline{C}^{-1}$ could directly be used. This possibility, developed below, presents the particularity that it makes a distance computed with k measurements, $k < N$, dependent on all initially available N measurements, with the risk of distorting the distance measure by biasing it towards nonexistent measurements.

Using this measure is equivalent to setting $\underline{m}_{n-k} = \underline{0}$ in (3.1-12), thus the equation becomes:

$$d_k'^2 = d_k^2 + ||(\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{X} \cdot \underline{L})^T||_2^2 \qquad (3.1\text{-}17)$$

and, in the case where $k = n-1$:

$$d_k'^2 = d_k^2 + \beta(\underline{m}_k^T \cdot \underline{C}_k^{-1} \cdot \underline{x})^2 \qquad (3.1\text{-}18)$$

These equations show the new measure, $d_k'^2$, to be always larger than the equivalent measure, d_k^2, which uses the true inverse of $\underline{C}_k$. Therefore, this new measure presents a double advantage: better separation and a single inverse matrix computation. It is noted, however, that this

computation may not always be possible, for reasons of computation time, storage memory, and rounding errors.

Unfortunately, this new measure does not increase monotonically as new components are added. Indeed, using the notation of (3.1-12), the relationship between the distances computed with n and with k components is as follows:

$$d'^2_n = d'^2_k + \underline{m}^T_{n-k} \cdot (2 \cdot \underline{V}^T \cdot \underline{m}_k + \underline{W} \cdot \underline{m}_{n-k}) \qquad (3.1\text{-}19)$$

and, in the case where $k = n-1$:

$$d'^2_n = d'^2_k + \mu \cdot (2 \cdot \underline{v}^T \cdot \underline{m}_k + \beta \cdot \mu) \qquad (3.1\text{-}20)$$

Thus added components may actually decrease the total distance and the maximum value is no longer necessarily reached by using all components. This measure then appears as less representative of the actual situation than the other; nevertheless, its computational advantages way outweigh this problem.

3.2. Geometric Interpretation

The results of the previous section can be better understood through the use of a geometric representation. In order to appreciate the full meaning of the latter, a few remarks are necessary.

It is known from matrix theory [18] that the quadric of equation:

$$\underline{x}^T \cdot \underline{A} \cdot \underline{x} = \text{constant} \qquad (3.2\text{-}1)$$

where $\underline{A}$ is positive definite, represents an ellipsoid. Moreover, the problem of minimizing $||\underline{x}||_2$ subject to the condition (3.2-1) is equivalent to that of maximizing $\underline{x}^T \cdot \underline{A} \cdot \underline{x}$ subject to the condition $||\underline{x}||_2 = \text{constant}$ [18, p. 405]. Finally, it is noted that, if $\underline{A} = \underline{C}^{-1}$, the variance in the direction of $\underline{x}$ is given by:

$$1/\sigma^2(\underline{C},\underline{x}) = \underline{x}^T \cdot \underline{C}^{-1} \cdot \underline{x} / \underline{x}^T \cdot \underline{x} \qquad (3.2\text{-}2)$$

This ratio is known as the Rayleigh quotient. Thus the two problems mentioned above are equivalent to maximizing the Rayleigh quotient, or minimizing the variance, along all axes.

This allows a geometric arrangement for interpreting the Mahalanobis distance: the quadric (3.2-1) is drawn for an arbitrary value of the constant, and a line segment drawn from the unknown vector to the mean of the class (this is the support of the vector $\underline{x}$ used above). The Mahalanobis distance is then directly proportional to the ratio of the Euclidean distance between the unknown vector and the mean of the class (i.e., $\sqrt{\underline{x}^T \cdot \underline{x}}$) to the radius of the quadric in this direction. (If the quadric is drawn for a constant value of 1, this radius is the standard deviation of the set in this direction, and the Mahalanobis distance is directly equal to the ratio defined above.) This is illustrated in Figure 3. By using this representation, it becomes possible to illustrate geometrically the results of section 3.1.

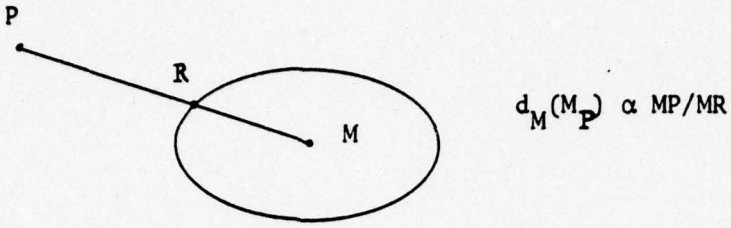


Figure 3. Illustration of the Mahalanobis distance.

First, consider Figure 4. The point M is the mean of the class, and the point P is the unknown pattern. The coordinates of these two points on both axes are denoted by a subscript x or y. The Mahalanobis distance using both x and y components is proportional to $d_E(M,P)/d_E(M,I)$; using only the x component, the distance is proportional to $d_E(M_x,P_x)/d_E(M_x,I_x)$, where I_x is not, in general, the x-coordinate of I. Since I_x is always situated between P_x and the projection of I on the x axis, elementary geometry insures that the measure using both components is always larger than that using only the x component. (Similar reasoning can be held for the y component.) Equality occurs when the principal axes of the ellipsoid are parallel to the coordinate axes and MP is along one of the axes (i.e., $\underline{C}_k$ is diagonal and $\underline{x}$ has only one nonzero component).

Then consider Figure 5. As the original covariance matrix is kept, the result is that the original quadric, not its projection, is used as a measure of spread. Clearly, the ratio $d_E(M,P_x)/d_E(M,I_x')$ is always larger than, or equal to, the ratio $d_E(M_x,P_x)/d_E(M_x,I_x)$; the same remark holds for measures along the y-axis. It is noted that the ratio $d_E(P_x,M)/d_E(M,I_x')$ is even larger than the ratio $d_E(P,M)/d_E(M,I)$,

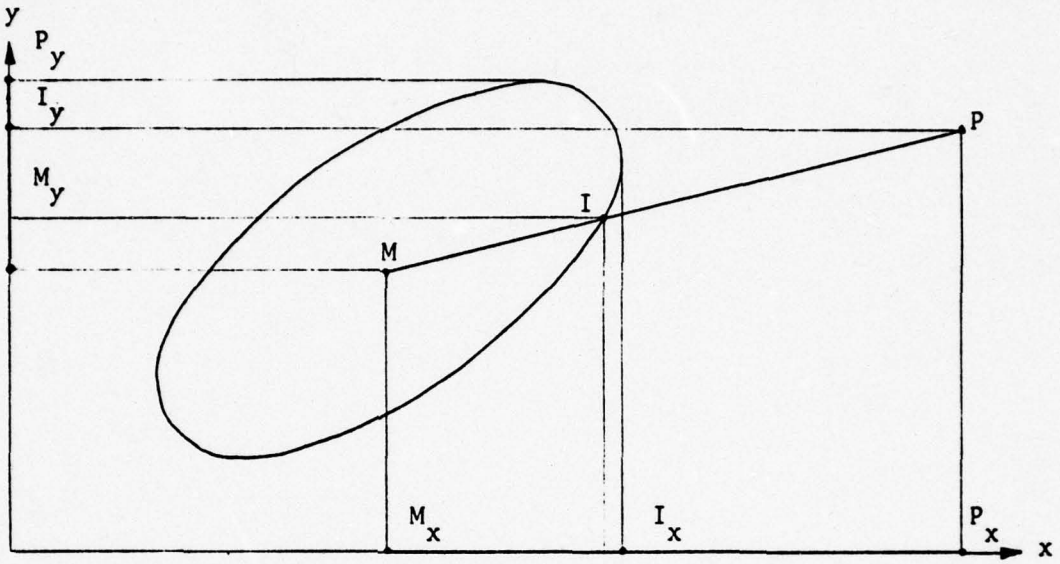


Figure 4. Illustration of the first type of dimensionality reduction.

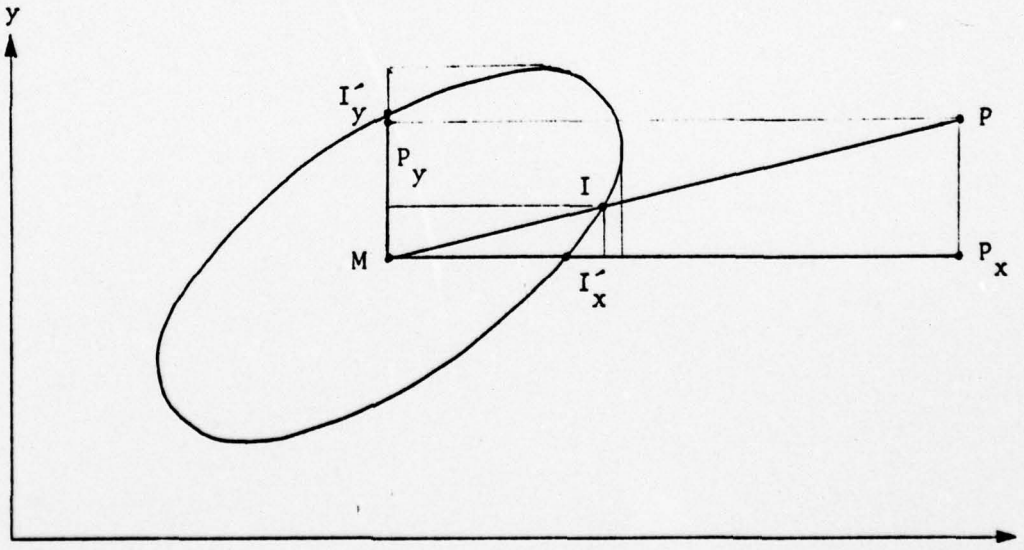


Figure 5. Illustration of the second type of dimensionality reduction.

meaning that the maximum of this second type of measure is not reached when using both components, but when using the x component alone.

However, in a multiclass case, an increased distance does not imply better classification performance, at least when using classifiers of the minimum distance type. This is due to the fact that only ratios change; the actual information available constantly decreases when components are removed.

CHAPTER 4

INTERACTIVE SELECTION PROCEDURE

4.1. Optimal and Suboptimal Searches

The problem of selecting the "best" subset of components is a search problem. That is to say, it requires the evaluation of each possible subset against the chosen criterion and then the choice of the most satisfactory subset. However, for a set of N measurements, there are 2^N-1 non empty subsets; thus the number of possible subsets gets rapidly out of hand. Even when the choice is limited to subsets of, say, k components, there still are $\binom{N}{k}$ possible subsets. To appreciate the size of these numbers, let N = 50 and k = 20; then 2^N-1 is of the order of 10^{15} and $\binom{N}{k}$ of the order of 10^{13}. These are clearly forbidding numbers; thus an exhaustive search is, for all practical purposes, impossible.

An attractive solution is the stepwise search, in which, at the k-th step, that of the remaining N-k components which, when added to the k already selected, yields the best subset of k+1, is selected. To obtain k components, this method will require $N + (N-1) + \ldots + (N-k) = N(k+1) - 1 - \frac{1}{2}k(k+3)$ searches. For the previous example, this yields 819 searches, which is well within feasible values. Unfortunately, such an approach does not, in general, yield the optimal solution. This is a typical feature of multivariate problems, where the interdependence of the various parameters requires that all of them be considered together. In the particular case where the variables are statistically

independent, a stepwise procedure will yield the optimal solution; otherwise, nothing can be said about the value of the selected subset. (This is true for the selection problem, but not for the feature ordering problem—as encountered in sequential pattern recognition—in which case some optimality properties can be demonstrated [13]). Nevertheless, stepwise procedures (also called sequential procedures) constitute the only tools available for deterministic selections.

A sequential selection procedure can start at both ends; that is, it can either add the next best, or delete the next poorest, measurement. These two methods are referred to as forward and backward sequential searches. Stopping criteria must be provided for automated search procedures to avoid ending with a trivial selection (all components or none at all). Such criteria are chosen in accordance with the selection strategy to determine the significance of prospective new measurements. Equipped with well-chosen criteria, sequential methods will find suboptimal solutions.

All selection procedures, however, are based on statistical strategies. The structural information which may be available is not taken into consideration. Other choice criteria, such as cost of measurement, sensitivity to noise, or reliability of individual measurements, are difficult to incorporate, despite some efforts in the field of dynamic programming. In general, the above-mentioned information, structural in particular, cannot very well be fed to computers. Human interpretation is necessary at this level; this is the rationale behind interactive procedures.

Therefore, it seems desirable to develop a procedure which provides a human user with all the facilities of automated search methods and accepts human input to direct the use of searches.

4.2. Computational Problems

As stressed before, the computation of the inverse matrix, $\underline{C}^{-1}$, is a delicate task when the dimension of $\underline{C}$ is large. This problem is further complicated by the requirement that $\underline{C}^{-1}$ be positive definite despite the rounding errors. Unfortunately, sample covariance matrices are often ill-conditioned, due to the lack of sample vectors. (In this respect, $\underline{W}$—as defined by (2.3-3)—is likely to present a better behavior.)

The best way to minimize rounding errors is to use the Cholesky decomposition; this decomposition is very stable from a numerical point of view. The quadratic form can then be rewritten:

$$d^2 = \underline{x}^T \cdot \underline{C}^{-1} \cdot \underline{x} = \underline{x}^T (\underline{L} \cdot \underline{L}^T) \tilde{\underline{x}}^1 = ||(\underline{L}^T)^{-1} \cdot \underline{x}||_2^2 \qquad (4.2\text{-}1)$$

which can also be written:

$$d^2 = ||\underline{y}||_2^2 \quad , \quad \underline{L}^T \cdot \underline{y} = \underline{x} \qquad (4.2\text{-}2)$$

As $\underline{L}$ is lower triangular, $\underline{y}$ can be immediately found by back-substitution This method has the added advantage of insuring that the quadratic form will be nonnegative. Unfortunately, it is extremely difficult to introduce a recurrence (of the type developed in section 3.1) in these formulas.

It is noted, however, that the updating procedure summarized by equations (3.1-9) is very stable and presents the advantage of determining if the updated inverse is positive definite (thanks to the test on the value of β). Thus the selection process will not be more sensitive to the condition of the covariance matrices than direct inversion procedures.

Inverting the matrices directly might still be desirable; since the estimation of the distance requires more operations using (4.2-2) than by straight matrix multiplication using the inverse matrix, computation time might be reduced by using inverses. In such a case, it usually proves better to invert the triangular matrix, $\underline{L}^T$, than the original matrix ([6],[22]). (This can be related to the fact that the condition number of $\underline{C}$ is the square of that of $\underline{L}^T$.) Nevertheless, negative values might be obtained when computing distances. When using these distances for classification, however, the problem is lessened since the error introduced in the measure is approximately constant and thus, within reasonable limits, should not change the classification.

4.3. Set-Up of the Procedures

Several procedures have been programmed to implement interactive sequential selection, making use of the results of sections 2.3 and 3.1. Both measures of average interclass distance—as discussed in section 2.3—can be used; it was felt that the arithmetic average of all interclass distances could prove useful as, with the same set-up, selection criteria could be directed towards various subsets of these distances (an unrealizable option while using the generalized measure).

Programs were designed to use the updating procedures described in section 3.1. An additional procedure, using the arithmetic mean criterion and working with the large inverse matrices, was programmed, to be used with sets of reduced dimension. All procedures can perform forward and backward sequential searches—though the backward search can only start from the point reached by the forward search. Variable thresholds are used as stopping criteria, either in absolute or in relative value. When using the arithmetic mean criterion, weights can be assigned to specific distances in order to bias the search; moreover, the search can be directed to try to isolate a specific class or to use only the distance between a specified pair of classes.

The results of the selection are judged by attempting to classify the training patterns (with and without a jackknife) and patterns generated by slight modifications to the training set. It will be recalled that the jackknife procedure consists of leaving some patterns out of the training set, classifying them with the selection obtained by using the diminished training set, and repeating the procedure until all patterns have been used. In the absence of real data, the jackknife method allows a more realistic estimation of the performance of a recognition system; however, it is a time-consuming procedure.

The classifier used was chosen to be well suited to the selection criteria, simple (since the purpose of the experiments was selection and not classification), and still able to produce good results. An obvious choice was the minimum distance classifier (using the Mahalanobis distance), which assigns a pattern to the class—represented by its mean—closest to it.

The programs were written in FORTRAN V and run on the DEC-system 10 computer of the University of Tennessee, with an allocation of 65 k words of main memory.

CHAPTER 5

MOMENTS AS FEATURES

As mentioned in Chapter 1, feature extraction is the first stage in a pattern recognition process. Thus, in order to test the selection procedures developed in this study on real data, some method of feature extraction must be chosen. In the absence of any information concerning the data, global feature descriptors are more desirable than local ones; also, some properties of invariance to several types of perturbation should be found in any good feature descriptor.

Moments satisfy all these requirements. They have been extensively used in classical mechanics and statistics, and several researchers have used them for pattern recognition purposes ([11], [16]). It will be recalled that the moment of order p of a discrete function f(x) over a finite interval is defined by:

$$m_p = \sum_x x^p \cdot f(x) \tag{5-1}$$

where the summation is taken over all x in the interval. This is easily extended to n-dimensional moments as follows. The $p_1 p_2 \ldots p_n$-th moment (of order $p_1 + p_2 + \ldots + p_n$) of the function $f(x_1, x_2, \ldots x_n)$ is defined as:

$$m_{p_1 p_2 \cdots p_n} = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_n} x_1^{p_1} \cdot x_2^{p_2} \cdots x_n^{p_n} \cdot f(x_1, x_2, \ldots x_n) \tag{5-2}$$

M. K. Hu [16] has noted that two-dimensional moments of a continuous function—the summations are then replaced by integrals—will, under some finiteness assumptions, uniquely determine the function itself and, vice versa, the function uniquely determines the sequence of moments. In a discrete case, it is logical to assume that only a finite set of moments is necessary to represent a function over a finite interval. The following proof is due to Barrero [3] and restricted, for simplicity, to a one-dimensional case.

Let the discrete function f of a discrete variable x_i be defined over the interval i = 1, ..., N; this function can then be represented in vector form as follows:

$$\underline{f} = [f(x_1)\ f(x_2)\ \ldots\ f(x_n)]^T \tag{5-3}$$

Let the sequence of moments of order p, p = 0, ..., N - 1, also be represented by a vector:

$$\underline{m} = [m_0(F)\ m_1(F)\ \ldots\ m_{N-1}(F)] \tag{5-4}$$

Then a matrix $\underline{X}$ can be defined such that:

$$\underline{m} = \underline{X} \cdot \underline{f} \tag{5-5}$$

and this matrix reads as follows:

$$\underline{X} = \begin{bmatrix} x_1^0 & \cdots & x_i^0 & \cdots & x_N^0 \\ x_1^1 & \cdots & x_i^1 & \cdots & x_N^1 \\ \vdots & & \vdots & & \vdots \\ x_1^{N-1} & \cdots & x_i^{N-1} & \cdots & x_N^{N-1} \end{bmatrix} \tag{5-6}$$

If $x_i \neq x_j$, $i \neq j$, this matrix is nonsingular (its determinant is known in matrix algebra as a Vandermonde determinant [5], [18]). Thus it has an inverse, $\underline{X}^{-1}$, and the transformation can be reversed:

$$\underline{f} = \underline{X}^{-1} \cdot \underline{m} \tag{5-7}$$

which proves that N moments are sufficient to represent uniquely N samples of a discrete function. It is further noted that the use of x in the above equations implies only that x be a single-valued function of the index i. This allows the use of various functions in order, for instance, to reduce the computational problems arising with high order moments, or to pick out specific regions in the interval.

Invariant properties of moments are easily developed. The following moments, called central moments, are invariant under translation:

$$\mu_p = \sum_x (x - \bar{x})^p \cdot f(x), \quad x \text{ in the interval} \tag{5-8}$$

where $\bar{x}$ ($= m_1/m_0$) corresponds to the center of mass of the function. In order to achieve invariance to size, these moments are normalized as follows:

$$\eta_p = \mu_p / m_0^{(p/2 + 1)} \tag{5-9}$$

In particular, $\eta_0 = 1$, $\eta_1 = 0$.

It can be further demonstrated [16] that the theory of algebraic invariants is applicable to moments, so that combinations of moments can be developed that are invariant under general linear (nonsingular) transformations.

Some moments are well known throughout the technical literature. The moment of order zero is the sum of all values of the function in the interval and thus represents the "mass" of the function. The noncentral moments of order one are related to the center of mass; the moments of order two are related to the principal components (central moments of order two express variances and covariances). Finally, moments of order three are associated with the skewness of the function and moments of order four with its kurtosis.

Thus moments appear as useful global feature descriptors; they are easily combined to form invariants; with the use of an appropriate index function, they can be made to extract local features. Finally, the computational problems associated with moments of large orders can be overcome by scaling methods or by choosing an index function with a reduced dynamic range (it is noted, however, that this last procedure is likely to diminish the discriminant power of the moments, and to make the matrix in (5-6) ill-conditioned).

CHAPTER 6

EXPERIMENTAL RESULTS

6.1. Iris Data

The first application example is based on the classic taxonomy problem presented by Fisher [12]. This problem involves three species of Iris, namely I. setosa, I. versicolor, and I. virginica, on which four measurements have been performed: sepal length, sepal width, petal length, and petal width. The class of I. setosa is known to be linearly separable from the two others, which present considerable overlap. This is obviously a rather simple case, which can be investigated by exhaustive search; this, and the fact that numerous researchers have studied the problem from various points of view, make the Iris data a useful basis of comparison.

Fisher's original data consist of fifty sets of measurements—fifty patterns—per class. Tests were run without jackknife procedure, using both criteria presented in section 2.3. The additional procedure using the complete inverse matrices was also used, since the small number of measurements would not allow severe distortions of the measure. For comparison purposes, exhaustive classification tests were run, using both the "true" inverse covariance matrix (i.e., the inverse of the reduced covariance matrix) and the inverse matrix subsets. First, a classification test was run using all four measurements to establish a basis for comparison. The resulting classification matrix is shown

in Table 1. It corresponds to a total misclassification rate of 2 percent. Using a similar classifier, the BMD program, P7M [10], presents the same error percentage, where a method based on density estimation [6] resulted in a perfect score.

Table 1. Iris classification matrix using all measurements.

from: to:	I. setosa	I. versicolor	I. virginica
I. setosa	50	0	0
I. versicolor	0	47	3
I. virginica	0	0	50

The selection program based on the large inverse matrices was then run using both forward and backward sequential procedures; these always converged to the same result. Thus subsets of three, two, and one measurements were determined, which resulted in misclassification rates of 2, 3-1/3, and 4-2/3 percent respectively. The classification matrices for these selections are shown in Table 2, where the measurements have been numbered for commodity (1: sepal length, 2: sepal width, 3: petal length, 4: petal width).

The exhaustive classification tests using the large inverse matrices showed all selections to be indeed optimal. A second subset of three measurements and a second subset of two measurements demonstrated the same error percentage as the selected subsets, but none better could be found. It is interesting to note that twelve of the fourteen possible

Table 2. Iris classification matrices (I) using (a) measurements 2, 3, 4; (b) measurements 3, 4; (c) measurement 3.

from: to:	I. setosa	I. versicolor	I. virginica
	(a) Measurements 2, 3, 4		
I. setosa	50	0	0
I. versicolor	0	48	2
I. virginica	0	1	49
	(b) Measurements 3, 4		
I. setosa	50	0	0
I. versicolor	0	47	3
I. virginica	0	2	48
	(c) Measurement 3		
I. setosa	50	0	0
I. versicolor	0	46	4
I. virginica	0	3	47

subsets correctly classified all patterns of the first class; only the first or the second measure taken alone resulted in a few errors for this class (while the same subsets completely confused I. versicolor and I. virginica). The classification results are summarized in Table 3.

Table 3. Error rates associated with Iris measurement subsets (I).

Subset	Error Rate	Subset	Error Rate
1,2	18	2,3	4-2/3
1,3	3-1/3	2,4	4-2/3
1,4	8	3,4	3-1/3
1,2,3	4	1	28
1,2,4	8	2	38
1,3,4	2	3	4-2/3
2,3,4	2	4	5-1/3

The same procedure was repeated with the other two procedures, which use the updating method to compute the "true" inverse at each step. The generalized distance criterion led to rather poor selections (with 4, 4, and 28 percent of error respectively), while the arithmetic mean criterion resulted in the optimal selection for each subset dimension. Classification matrices are shown in Table 4, and the exhaustive classification results summarized in Table 5. It is noted that a better classification was achieved by using three measurements than by using all four of them. The comparison between the two approaches (subsets of the large inverse matrix versus "true" inverse matrix) confirms what has been said in section 3.1: the classification results are better

Table 4. Iris classification matrices (II) using (a) measurements 1, 3, 4; (b) measurements 3, 4; (c) measurement 4.

from: to:	I. setosa	I. versicolor	I. virginica
	(a) Measurements 1, 3, 4		
I. setosa	50	0	0
I. versicolor	0	48	2
I. virginica	0	0	50
	(b) Measurements 3, 4		
I. setosa	50	0	0
I. versicolor	0	47	3
I. virginica	0	1	49
	(c) Measurement 4		
I. setosa	50	0	0
I. versicolor	0	48	2
I. virginica	0	4	46

Table 5. Error rates associated with Iris measurement subsets (II).

Subset	Error Rate	Subset	Error Rate
1,2	19-1/3	2,3	4
1,3	4	2,4	5-1/3
1,4	3-1/3	3,4	2-2/3
1,2,3	4	1	28
1,2,4	4-2/3	2	44-2/3
1,3,4	1-1/3	3	4-2/3
2,3,4	3-1/3	4	4

with the "true" inverse. Moreover, the selection operated by one method is not optimal when used with the other method. Both methods, however, achieved optimal solutions; while this cannot be expected to remain true for more complex problems, it is a proof that sequential procedures are able to choose better-than-average subsets.

6.2. Radar Returns of Aircraft

In this experiment, eight different aircraft were used, four American planes and four Russian planes. These aircraft had been selected for a previous investigation [15] because they form four more or less similar pairs: two large transports, two light bireactor transports, two large fighters, and two small fighters. A list of the eight planes with their wingspan is given in Table 6.

The radar returns were carefully simulated by M. Harris [15], using model airplanes. As used in the following experiments, a radar image consists of a set of reflection points, characterized by their coordinates on a 512 by 512 grid, around which electromagnetic scatter is simulated

Table 6. Aircraft used in the experiments with radar returns.

Name	Type	Nationality	Wingspan
Mikoyan MIG-21	Fighter	USSR	7.15 m
McDonnell-Douglas A4M	Fighter	USA	8.38 m
Mikoyan MIG-25	Fighter	USSR	12.19 m
General Dynamics FB-111A	Fighter	USA	19.20 m
Tupolev TU-22	Transport	USSR	27.70 m
Boeing 737	Transport	USA	28.35 m
Antonov AN-22	Transport	USSR	64.39 m
Lockheed C-5A	Transport	USA	67.88 m

by generating a cluster of points with a two-dimensional Gaussian random generator. Typical images have from five to fifteen reflection points, each with a cluster of twenty to fifty points; one such image is shown in Figure 6.

Sixty-four patterns were generated for each class, for a total of 512 different images. The patterns correspond to about thirty different positions of each aircraft, each pattern thus obtained being used twice with the random generator. Figures 7 and 8 illustrate some of the patterns thus generated for the FB-111A and the B-737.

Central moments were used to describe the images. Thirty-three moments were computed, namely all moments of order two to order seven, using a reduced index described by the function:

$$x = 1 + i/(512 \cdot 10^{-2}) \quad , \quad i = 1, \ldots, 512 \qquad (6.2\text{-}1)$$

The moments were all divided by m_{00}, the zero-th order moment, but were not normalized since invariance to size was not desired. In fact, the

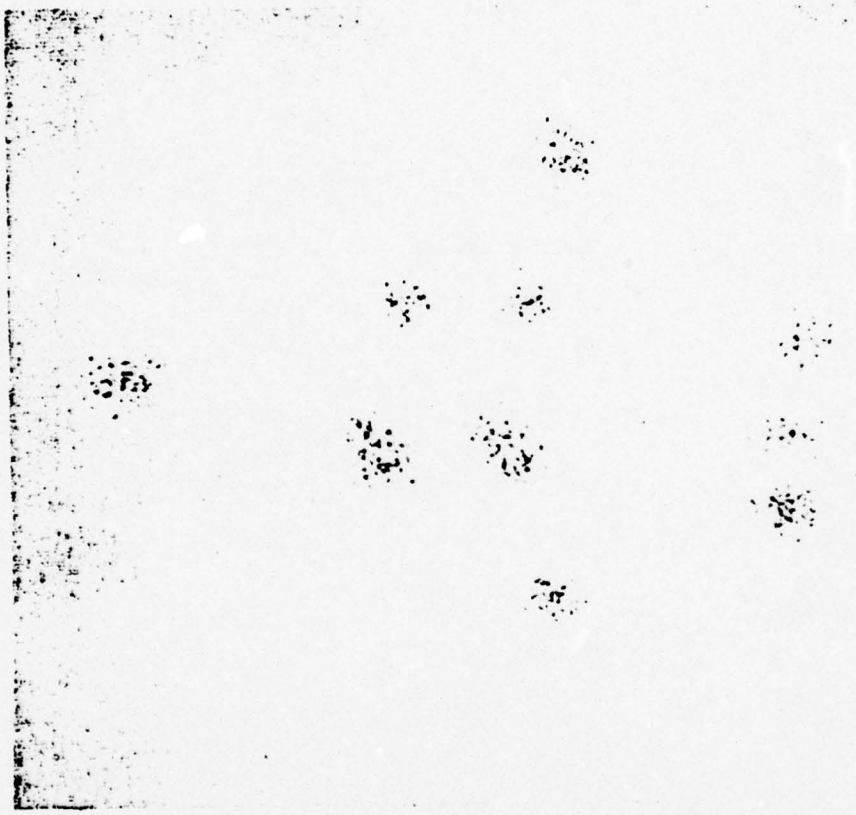


Figure 6. Typical simulated radar image.

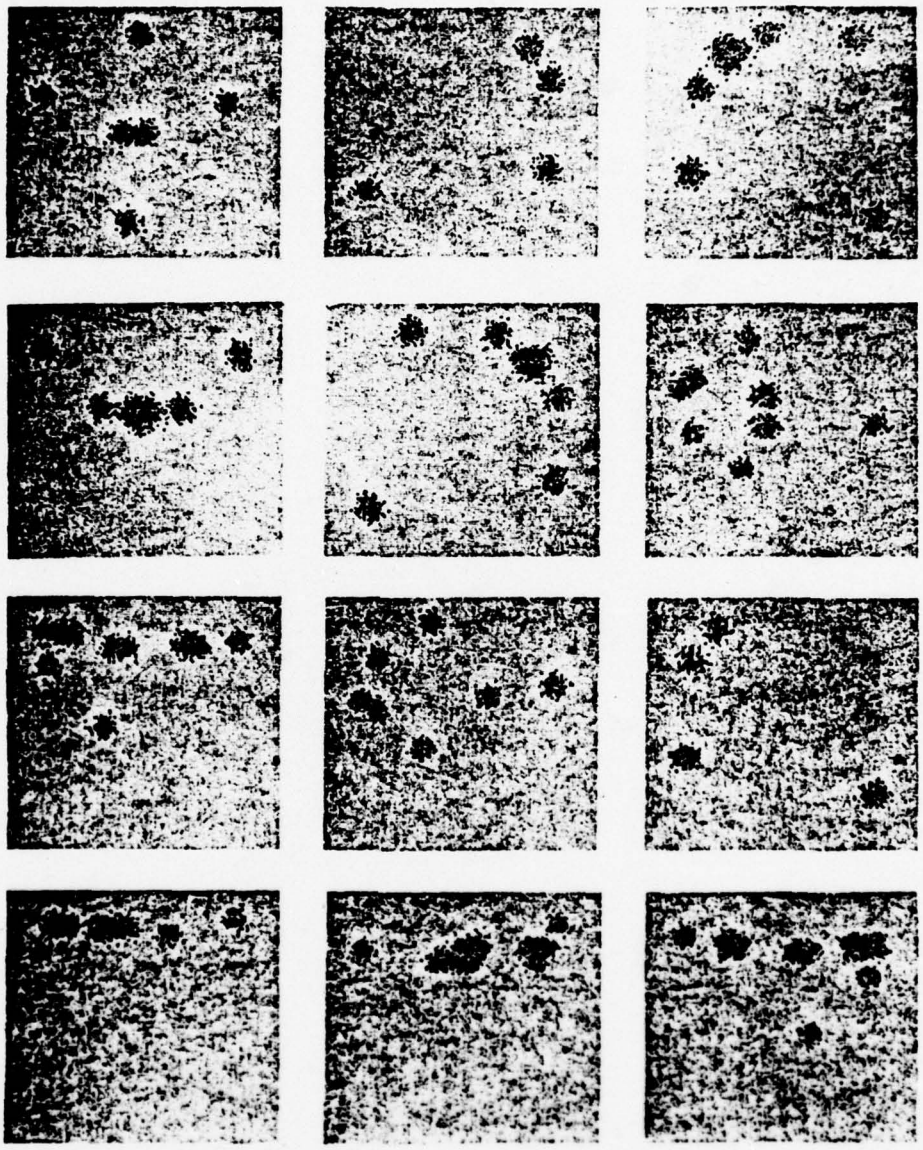


Figure 7. Sample patterns of the FB-111A class.

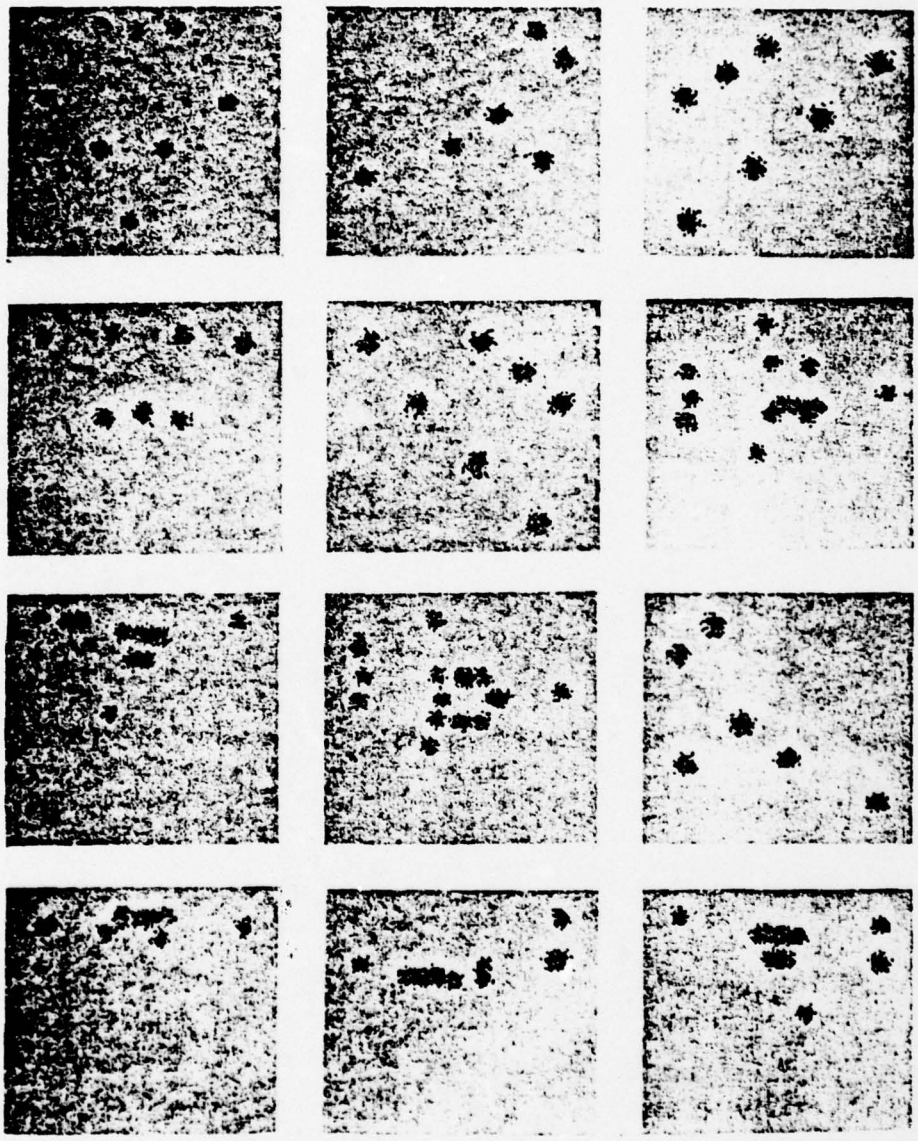


Figure 8. Sample patterns of the B-737 class.

size of the airplanes is a very important factor in this type of recognition problem. As radars usually provide range information, or even automatic scaling, the size can be taken into account irrespective of the distance at which the plane is flying. These experiments assume an automatic prescaling so that images are all of correct proportions. The scaling can also be accomplished with the moments themselves: the gyration radius of an image (i.e., $\sqrt{\mu_{02} + \mu_{20}}$) is known to be inversely proportional to the distance between the object and the imaging device and can thus be used together with the range information to scale the other moments [11].

Three series of experiments were conducted. In the first series, all eight airplanes were used, whereas the second and the third series dealt with the fighters and the transports respectively. As a preliminary step, classification tests were run using all thirty-three moments. The results were surprisingly good: only 6 out of the 512 patterns were misclassified, yielding an error rate of 1.17 percent. The classification matrix for this experiment is shown in Table 7. It is noted that there was not any confusion between fighters and transports.

Jackknife procedures resulted in an overall error rate of 2.9 percent (fifteen errors). Since the difference between both results was so small, and because jackknife classification required much more time than normal classification, the jackknife procedure was abandoned for the rest of the experiments. It is felt that the comparison between the performances of two subsets is as valid with normal evaluation methods as with jackknife procedures.

Table 7. Aircraft classification matrix using all 33 moments.

from: to:	MIG21	A4M	MIG25	FB111	TU22	B737	AN22	C5A
MIG21	63		1					
A4M	1	63						
MIG25			64					
FB111A			3	61				
TU22					64			
B737					1	63		
AN22							64	
C5A								64

The first test was designed to evaluate the usefulness of the large inverse matrix method as applied to high-dimensional data. A selection of 11 components was obtained using the arithmetic mean criterion, which yielded a catastrophic error rate of 80.2 percent. Directing the selection to use the smallest distance only as criterion, the result improved to 76.4 percent of error. Using the "true" inverse covariance matrix of each subset for classification diminished the error rates to 64.5 and 51.8 percent respectively. Most patterns were classified as C5A, i.e., attributed to the class which had the largest spread. Other selections confirmed these results, clearly showing the inapplicability of this procedure to high-dimensional data.

Random selections of 11 components were then performed and classification tests run to obtain a mean value of performance with which to compare the results of the first two selection procedures. For the classification among eight airplanes, an average error rate of 40 percent was observed, while the two-class fighter-transport

classification was correct 90 percent of the time. Both selection procedures were then used; both selected the same subset, for which classification tests showed an error rate of 17.2 percent for the eight-class problem and 2.7 percent for the two-class problem. The classification matrix for this experiment is given in Table 8.

Table 8. Aircraft classification matrix using 11 selected moments.

from: to:	MIG21	A4M	MIG25	FB111A	TU22	B737	AN22	C5A
MIG21	37	13	14					
A4M	3	50	11					
MIG25		2	53	1	7	1		
FB111A			12	46	1	5		
TU22					52	8		4
B737					5	58		1
AN22							64	
C5A								64

Various other subsets were selected using these two procedures; for each selected subset, several other subsets were randomly chosen (with the same number of components) to provide a basis for comparison. Some of these results are summarized in Table 9. It is noted that the selected subset always allowed much more accurate classifications. A second important feature is the relationship between the number of selected components and the error rate; Figure 9 illustrates this relationship. The curve presents two thresholds, between which the increase in performance is very slow; thus a practical pattern recognition system would use either a dozen measurements or all of them.

Table 9. Classification results for various subsets.

Number of Components	Selected Subset Error Rate		Random Subset Error Rate	
	8-class	2-class	8-class	2-class
1	62.6	31.2	81.	50.
3	38.3	16.7	70.	32.
5	37.1	15.2	54.	28.
9	16.1	6.5	45.	16.
11	17.2	2.7	40.	10.
18	18.1	3.	27.	7.
23	12.7	3.1	21.	6.

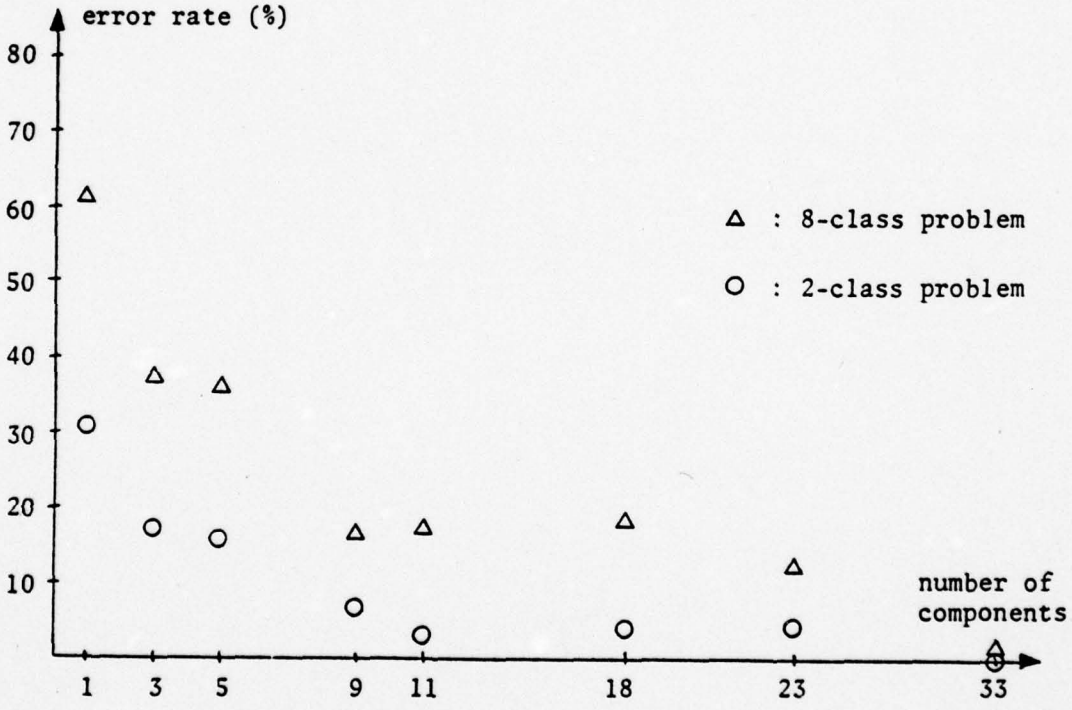


Figure 9. Relationship between the number of selected components and the error rate.

The second series of experiments used only the four fighters, since the distinction between fighters and transports seemed well-defined. Two selections were obtained, one of nine, and one of eighteen components. In agreement with the curve of Figure 9, the selection of nine measurements yielded slightly better results (9 percent versus 9.4 percent of error). Classification matrices are shown in Table 10. It is noted that the first seven moments selected were the same as those selected for the 8-class problem. (The same seven moments will also be the first selected in the next series of experiments.)

Table 10. Fighter aircraft classification matrices using (a) nine selected moments and (b) eighteen selected moments.

from: to:	MIG21	A4M	MIG25	FB111A	MIG21	A4M	MIG25	FB111A
MIG21	45	7	12		47	7	10	
A4M		63	1		1	61	2	
MIG25		1	63			1	63	
FB111A			2	62			3	61

The last series of tests was a repetition of the second, this time using the four transports. As expected, there were less misclassifications with these planes as the selected subsets resulted in error rates of 4.7 and 7.8 percent for nine and eighteen components respectively. The classification matrices are shown in Table 11.

These three series of experiments have shown both the discriminating power of moments and of the minimum Mahalanobis-distance classifier and the usefulness of sequential search procedures. Selected subsets

Table 11. Transport aircraft classification matrices using (a) nine selected moments and (b) eighteen selected moments.

from: to:	TU22	B737	AN22	C5A	TU22	B737	AN22	C5A
TU22	55	8		1	57	7		
B737	1	62		1	13	51		
AN22			63	1			64	
C5A				64				64

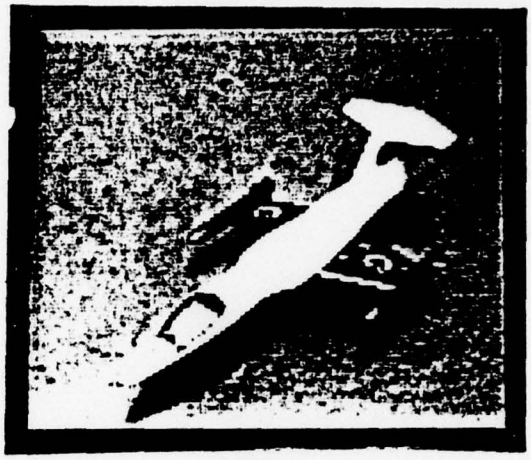
resulted in two to three times less misclassifications than equivalent subsets chosen at random. Moreover, the selection allowed a reduction of a factor three in the number of measurements for a decrease in performance of about 10 percent only.

6.3. Photographs of Aircraft

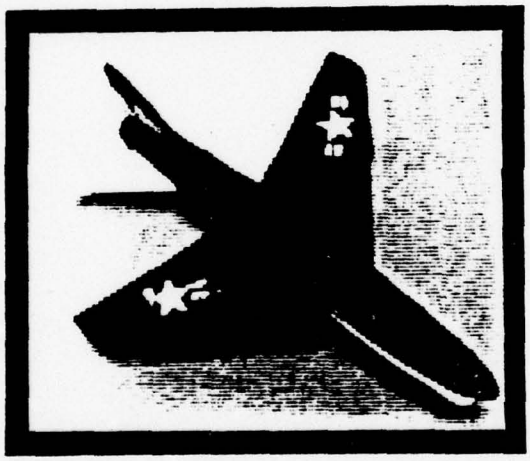
In this experiment, a series of four pictures was taken for each of four different toy aircraft. All four planes were fighters: the A4M "Skyhawk," the F104, the A7D "Corsair," and the Mirage F1. Each photograph represents a plane in a different position. Typical photographs are shown in Figure 10. The pictures were digitized in a grid of 128 by 128 points, using 256 levels of gray. Each picture was then numerically rotated 90, 180, and 270 degrees, and symmetrized around both vertical and horizontal axes, to yield a total of twenty-four images per airplane.

Six moment invariants were used to describe these pictures. (The choice of such a small number of features was based on the fact that photographs convey a large amount of information about aircraft—much

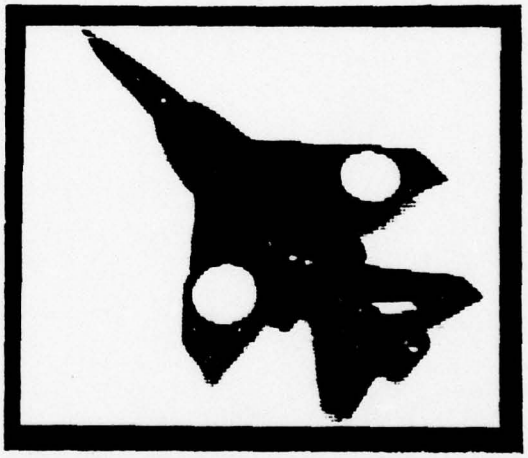
(a) A4M "Skyhawk"



(b) F104



(c) A7D "Corsair"



(d) Mirage F1

Figure 10. Sample photographs of the four aircraft.

more, for instance, than radar returns.) These invariants were derived from the central normalized moments of order two and three using formulas developed by M. K. Hu [16], and adopting a normalization by the gyration radius introduced by Dudani [11]. These invariants can be shown to remain unchanged under general linear transformations.

In terms of the central normalized moments, the invariants can be expressed as follows:

$$I_1 = [(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2] \; / \; (\eta_{02} + \eta_{20})^2$$

$$I_2 = [(\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2] \; / \; (\eta_{02} + \eta_{20})^3$$

$$I_3 = [(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2] \; / \; (\eta_{02} + \eta_{20})^3$$

$$I_4 = \{(\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$
$$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\} \; /$$
$$(\eta_{02} + \eta_{20})^6$$

$$I_5 = \{(\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$+ 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\} \; / \; (\eta_{02} + \eta_{20})^4$$

$$I_6 = \{(3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$$
$$- (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\} \; /$$
$$(\eta_{02} + \eta_{20})^6$$

(6.3-1)

The last invariant, I_6, is taken in absolute value, since it changes sign under improper transformations.

The computed invariants differed only in the fourth significant digit; such accuracy is due to the absence of digitizing errors in the rotations and symmetries, and cannot be expected for more general transformations. This result shows, however, that moment invariants can be easily and accurately computed.

Preliminary classification tests were run using all six measurements: they yielded a perfect score of 96 correct classifications. Both procedures were then used to select subsets of one, two, three, and four components, which were compared against randomly selected subsets. All subsets of four components were found to result in perfect classification, thereby confirming the assertion made above concerning the informational content of a photograph.

Subsets of three components as selected by both procedures yielded error rates of 31 percent and 17 percent. Typical random subsets resulted in an error rate of 25 percent; one subset, however, gave rise to a perfect classification (first three components). This exemplifies the fact that sequential methods do not insure an optimal solution; indeed some selections might be worse than trivial random ones, as was the case with the above-mentioned subset of three components.

While searching for subsets of two components, it was noticed that the interclass distance measure fell sharply between three and two components. Thus, such subsets were not expected to result in satisfactory performances. Indeed, classification tests demonstrated an error

rate of 44 and 50 percent for both selected subsets, and 60 percent for random selections.

The usefulness of interactive procedures was demonstrated by the discovery of the optimal subset of three components: this happened while trying to keep the simplest invariants, and directing the searches in consequence. It is also noted that, in this experiment as in the other two, the selection procedure based on the arithmetic mean criterion outperformed that based on the Wilks' measure criterion, although it required more time and more attention from the user.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

The development of two types of sequential feature selection procedures has been presented along with a new inverse matrix updating algorithm. These procedures have been applied to three different classification problems, two of which make use of moments as pattern features.

The first experiment demonstrated the ability of the procedures to find suboptimum—even optimum—measurement subsets. The second experiment verified this property by comparing the selected subsets with randomly chosen ones; it also evidenced the discriminatory power of moments. The third experiment demonstrated the invariant properties of moments and proved the usefulness of interactive procedures.

Further experiments should be performed using known structural information as features, in order to determine the extent to which the interactive capabilities of the procedures improve the selection. Research should also be carried on to reduce the computational errors, especially to establish bounds on the final resulting errors as related to the condition of the covariance matrix.

REFERENCES

1. Anderson, T. W., Introduction to Multivariate Analysis, John Wiley & Sons, New York, 1958.

2. _____, and R. R. Bahadur, "Classification into Two Multivariate Normal Distributions with Different Covariance Matrices," Annals of Mathematical Statistics, vol. 33, pp. 420-431, 1962.

3. Barrero, A., "Existence of a Moment Transformation," Private Communication.

4. Bartlett, M. S., "An Inverse Matrix Adjustment Arising in Discriminant Analysis," Annals of Mathematical Statistics, vol. 22, pp. 107-111, 1951.

5. Bellman, R. E., Introduction to Matrix Analysis, McGraw-Hill, New York, 1970.

6. Bryant, T. K., "Classification and Clustering Using Density Estimation," Ph.D. Dissertation, University of Missouri, Columbia, 1971.

7. Chang, T. K., and S. T. Dwyer III, "New Approach to Measurement Selection in Pattern Recognition," Proceedings of the Second International Joint Conference on Pattern Recognition, pp. 35-40, 1974.

8. Das Gupta, S., "Non-Singularity of the Sample Covariance Matrix," Sankhya A, vol. 33, pp. 475-478, 1971.

9. _____, "Classification Procedures: A Review," in Discriminant Analysis and Applications, T. Cacoullos, ed., Academic Press, New York, 1973.

10. Dixon, W. J., ed., BMD: Biomedical Computer Programs, University of California Press, Berkeley, 1975.

11. Dudani, S. A., et al., "Aircraft Identification by Moment Invariants," IEEE Transactions on Computers, vol. C-26, no. 1, pp. 39-45, January 1977.

12. Fisher, R. A., "The Use of Multiple Measurements in Taxonomic Problems," Annals of Eugenics, vol. 8, pp. 179-188, 1936.

13. Fu, K. S., Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968.

14. Gonzalez, R. C., and P. A. Wintz, Digital Image Processing, Addison-Wesley, Reading, Mass., 1977.

15. Harris, J. M., "Feature Extraction and Recognition of Two-Dimensional Data by the Method of Moments," M.S. Thesis, University of Tennessee, Knoxville, 1976.

16. Hu, M. K., "Visual Pattern Recognition by Moment Invariants," IRE Transactions on Information Theory, vol. IT-8, pp. 179-187, 1962.

17. Mahalanobis, P. C., "Historical Note on the D^2-Statistic," Sankhyā, vol. 9, pp. 237-240, 1949.

18. Noble, B., Applied Linear Algebra, Prentice-Hall, Englewood Cliffs, N.J., 1969.

19. Patrick, E. A., Fundamentals of Pattern Recognition, Prentice-Hall, Englewood Cliffs, N.J., 1972.

20. Rao, C. R., Linear Statistical Inference and Its Applications, John Wiley & Sons, New York, 1968.

21. Tou, J. T., and R. C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, Mass., 1974.

22. Wilkinson, J. H., and C. Reinsch, Linear Algebra, Springer Verlag, New York, 1971.

23. Young, T. Y., and T. W. Calvert, Classification, Estimation and Pattern Recognition, American Elsevier, New York, 1974.

APPENDIX A

CIRCULANT MATRICES AND THE MAHALANOBIS DISTANCE

Circulant matrices have been used in digital image processing [14] to reduce the computational work involved in image restoration. A circulant matrix is such that any of its rows is obtained by a circular shift of the previous row (considering the last row as being before the first):

$$\underline{C} = \begin{bmatrix} C_0 & C_1 & \cdots & C_{N-1} \\ C_{N-1} & C_0 & \cdots & C_{N-2} \\ C_{N-2} & C_{N-1} & \cdots & C_{N-3} \\ \vdots & \vdots & & \vdots \\ C_1 & C_2 & \cdots & C_0 \end{bmatrix} \tag{A-1}$$

These matrices have the remarkable property that they can be diagonalized by a two-dimensional Fourier transformation:

$$\underline{C} = \underline{F}^{-1} \cdot \underline{\Lambda} \cdot \underline{F} \tag{A-2}$$

where $\underline{F}$ denotes the forward Fourier transformation matrix. As Fourier matrices are unitary, $\underline{F}^{-1} = \underline{F}^{H}$ (where $\underline{F}^{H}$ denotes the Hermitean transpose of $\underline{F}$), and equation (A-2) becomes:

$$\underline{C} = \underline{F}^{H} \cdot \underline{\Lambda} \cdot \underline{F} \tag{A-3}$$

Thus the inverse of $\underline{C}$, $\underline{C}^{-1}$, can be written:

$$\underline{C}^{-1} = \underline{F}^{H} \cdot \underline{\Lambda}^{-1} \cdot \underline{F} \tag{A-4}$$

Introducing this in the expression of a quadratic form yields:

$$\underline{m}^{T} \cdot \underline{C}^{-1} \cdot \underline{m} = \underline{m}^{T} \cdot \underline{F}^{H} \cdot \underline{\Lambda}^{-1} \cdot \underline{F} \cdot \underline{m} = (\underline{F} \cdot \underline{m})^{H} \cdot \underline{\Lambda}^{-1} \cdot \underline{F} \cdot \underline{m} \tag{A-5}$$

It is noted that $\underline{F} \cdot \underline{m}$ is the two-dimensional forward Fourier transform of $\underline{m}$; moreover, each diagonal element of $\underline{\Lambda}$ is the one-dimensional backward transform of the corresponding row of $\underline{C}$ [5]. The computation of the quadratic form thus requires a series of Fourier transformations —which can be rapidly performed by an FFT algorithm—and a vector innerproduct, instead of a matrix inversion and a matrix multiplication followed by a vector innerproduct. As there exist dedicated hardware circuits to compute fast Fourier transforms, the gain in computation time over a conventional method can be quite substantial.

However, this approach is not suitable for recursive computations such as used in the procedures developed in this thesis. Moreover, it appears unlikely that covariance matrices present a circulant structure, as this would reduce the number of different elements of the matrix to N/2.

APPENDIX B

PROGRAM DESCRIPTIONS

There are four basic programs. Two of these perform the selection, one is used to invert the appropriate covariance submatrices, and the last classifies the training set using the inverses previously computed. Another program computes the covariance matrices and mean vectors from the training sets; still other programs are used in conjunction with each experiment to prepare training samples. Each program is described in some detail in the following pages. The order in which programs are surveyed corresponds to the order in which they are used in practice, from the preparation of sample patterns to the classification test. It must be kept in mind that these programs have been written for a 36-bit machine with a limited allocation of main memory (65k words). It is finally noted that the execution times given for each program are only indicative, since they are highly dependent on the amount of data to be processed.

B.1. Program PATGEN.FOR

This program is used to generate simulated radar patterns. It reads (from a specified disk file) a specified number of images; each image is a list of coordinate pairs, locating reflection points on a 512 by 512 grid, and ending with a terminator pair. Each image is then symmetrized around vertical and horizontal axes, and around the first bisector, to yield three more images. A random number generator is

used to generate scatter points around each reflection point; the number of scatter points is specified by the user. (The final number may be somewhat smaller, due to the elimination of identical pairs.) The resulting patterns are written to a specified disk file; each pattern is a list of coordinate pairs ending with a terminator. A typical application used 8 original images, each formed of about a dozen reflection points, to generate 64 patterns (as each image is used twice with the random number generator); each pattern consisted of about a thousand coordinate pairs; the CPU time—including the user dialogue—was around 40 seconds (the length is mainly due to the elimination of redundant coordinate pairs). The Gaussian generator is one-dimensional, giving rise to spherical clusters of scatter points around each reflection point.

B.2. Program MOMENT.FOR

This program is used to compute moments from simulated radar patterns. It processes several classes at once. Each class is represented by a disk file in which radar patterns of an aircraft are stored (as created by the program PATGEN.FOR). Central moments are computed, up to a specified order, and normalized by the zero-th order moment. The index function used in the computation is $X = 1 + I/(512 \cdot \alpha)$, where the constant α is specified by the user. When all moments have been computed for a given class, they are written to a specified disk file; each moment vector (i.e., the moments of a pattern) is written in turn, beginning with the low order moments, and varying the order in the

second coordinate faster than that in the first coordinate. A typical application involved 8 classes of 64 patterns each, using moments up to order 7 (i.e., 33 moments) and required a little less than 2 minutes of CPU time.

B.3. Program MOMINV.FOR

This program is used to compute six general moment invariants from digitized pictures. Each picture is a 128 by 128 array of integers in the range 0-255. Four integers are stored in each word in the disk files. The program reads a specified number of pictures, each from a different disk file. Pictures are read in column by column; after each new column has been read, a call is made to a routine, written in DEC-10 assembler code, to unpack the 4 bytes. Each picture is then rotated 90, 180, and 270 degrees, and symmetrized around both vertical and horizontal axes (all of these operations are performed by simple reordering of the picture elements), to yield 5 new pictures. Central normalized moments of order two and three are then computed for each picture, and combined to form the six invariants described in section 6.3. For reasons of storage space, each picture is processed individually and written to disk; this output file is unique and contains 6 moment vectors for each original picture. A typical application used 4 original pictures to yield 24 vectors of six invariants, and required a little more than four minutes of CPU time. (This stems from the necessity of processing $128^2 = 16384$ items of data for each picture.)

B.4. Program STAT.FOR

This program is used to compute the statistics of a training set, that is, the set mean vector and covariance matrix. Several sets can be processed at once. A set consists of a number of pattern vectors written without separation in a disk file. The program will read a set in main memory, compute—in double precision—its mean vector and covariance matrix, and write them in a specified disk file (one for each class). The mean vector is written first, followed by the covariance matrix, in full storage. The double precision is useful in reducing the rounding errors when inverting the covariance matrix. A typical application involved 8 classes, each with 64 patterns of 33 components, to yield 8 mean vectors and 33 by 33 covariance matrices, using less than a minute of CPU time (including user dialogue).

B.5. Program INVERT.FOR

This program is used to compute the inverse of selected covariance submatrices. Several classes can be processed at once; a class is represented by a disk file where class mean vector and covariance matrix are stored in double precision (as created by the program STAT.FOR). The user specifies the components to be used; only the corresponding submatrix is then inverted. The inversion routine uses a triangular decomposition with a column search procedure, as developed by J. K. Bryant [6], in order to minimize the rounding errors. When the submatrix of a class is inverted, the subset of the mean vector and the inverse submatrix are written—in single precision—to a specified file (one

per class). A message is issued if a matrix is found to be singular. No messages are printed if the inverse is detected as not positive-definite; the FORTRAN run-time library, however, will issue a warning stating an attempt to take the square root of a negative number. A typical application used 8 classes of 33 components and required less than 20 seconds of CPU time.

B.6. Program CLASS.FOR

This program is used to run classification tests. The user must specify a number of classes and give, for each class, the name of the disk file containing the training samples, the number of these samples, and the name of the disk file containing the inverse covariance (sub) matrix to be used. There exist two versions of the program: the first, CLASS1.FOR, requires that the number of components to be used be equal to the dimension of the inverse matrix, while the second, CLASS2.FOR, uses a submatrix of the inverse covariance matrix if the number of components to be used is less than the dimension of the inverse matrix. Both programs use only the components that the user specifies. The Mahalanobis distances from each training sample to each class are computed and stored in main memory. The computations, for reasons of storage space, are performed on a class per class basis: the mean vector and inverse covariance matrix of a class are read, and all training samples processed with this class. The programs print an extensive list, class per class, of all distances, with the result of the classification in the margin; in addition, the number of correctly

and incorrectly classified patterns is printed for each class. A typical application involved 8 classes of 64 training patterns each; using all 33 components resulted in a CPU time occupation of a little more than two minutes. The program CLASS1.FOR is intended to be used with the selection programs SELEC1.FOR and SELEC2.FOR, while the program CLASS2.FOR should be used in conjunction with the selection program SELEC3.FOR.

B.7. Program SELEC1.FOR

This program is used to select subsets of measurements using Wilks' matrices and the generalized Mahalanobis interclass distance measure as selection criterion. To initialize the program, the user must specify the number of classes, the total number of components, and, for each class, the name of the file where the mean vector and covariance matrix are stored (as created by the program STAT.FOR) and the number of patterns used to compute these statistics. The program reads, and stores in main memory, all mean vectors, and adds all covariance matrices (weighed by the number of patterns used to compute them) into a single matrix, the total within-class scatter matrix. Once the initialization is complete, the user has a choice of eleven commands at his disposition, namely:

1. Reject all components (partial initialization).
2. Add a specified component.
3. Reject a specified component.
4. Select the next best component(s) (unless its contribution is below the current threshold-to-select).

5. Reject the next poorest component(s) (unless its contribution is above the current threshold-to-reject).
6. Set the threshold-to-select to a specified value (in percent of the total interclass distance measure).
7. Set the threshold-to-reject to a specified value.
8. Print a separation line (to isolate particular sections of the printed output).
9. Type (to the user's terminal) the current status (i.e., number of selected components, value of the interclass distance measure, and both threshold values).
10. Print the status.
11. End of program (return to the monitor).

Each selection, or rejection, is indicated to the user, together with the contribution of the component in cause; this information is also printed. It is thus possible to request a complete feature ordering by starting with no components and requesting the maximum number of steps on the fourth command. Each selection, or rejection is immediately followed by an updating of the interclass distance measure and of the current inverse matrix.

Using 65k words of main memory, this program can deal with problems involving several dozens classes and up to eighty components. Apart from the initialization, which may require a few seconds of CPU time, no command takes more than a tenth of a second of CPU time.

B.8. Program SELEC2.FOR

This program is used to select subsets of measurements using the mean of all interclass distances as selection criterion. To initialize the program, the user must specify the number of classes, the total number of components, and, for each class, the name of the file where the mean vector and covariance matrix are stored (as created by the program STAT.FOR) and the name of the file to which the selected subset's mean vector and inverse covariance matrix are to be written. The program reads, and stores in main memory, all mean vectors and covariance matrices. Once the initialization is complete, the user has a choice of twenty-one commands at his disposition, namely:

1. Print a separation line (to isolate particular sections of the printed output).
2. Reject all components, reset all weights to one (partial initialization).
3. Add a specified component.
4. Reject a specified component.
5. Select the next best component(s), using the average of all weighted interclass distances as criterion (unless the resulting contribution is below the current threshold-to-select).
6. Reject the next poorest component(s), using the same criterion as under 5 (unless the resulting contribution is above the current threshold-to-reject).
7. Select the next best component(s), using the average of the weighted distances from a specified class to all others as

criterion (unless the resulting contribution is below the current threshold-to-select).

8. Reject the next poorest component(s), using the same criterion as under 7 (unless the resulting contribution is above the current threshold-to-reject).
9. Select the next best component(s), using the distance from one specified class to another specified class as criterion (unless the resulting contribution is below the current threshold-to-select).
10. Reject the next poorest component(s), using the same criterion as under 9 (unless the resulting contribution is above the current threshold-to-reject).
11. Type (to the user's terminal) the distances between two specified classes.
12. Type (to the user's terminal) the list of all selected components.
13. Type (to the user's terminal) the current short status (i.e., the number of selected components, the unweighted average interclass distance, its standard deviation, the minimum interclass distance, and both threshold values).
14. Print the current short status, plus the list of all selected components.
15. Print the current short status, plus the list of all selected components, plus the list of all interclass distances and their respective weights.

16. Set the threshold-to-select to a specified value (an absolute measure).
17. Set the threshold-to-reject to a specified value.
18. Assign a specified weight to the distance from a specified class to another specified class.
19. Type (to the user's terminal) the weight of the distance from a specified class to another specified class.
20. Type (to the user's terminal) the minimum interclass distance—in harmonic mean—and both class numbers.
21. Write (to the disk files specified during the initialization) the current inverse covariance matrices to disk and return to the monitor (end of program).

Each selection, or rejection, is indicated to the user, together with the weighted contribution of the component in cause; this information is also printed. Each selection, or rejection, is immediately followed by an updating of all interclass distances and all current inverse matrices.

Using 65k words of main memory, this program can deal with problems involving, for instance, eight classes of 33 components. Apart from the initialization, which may require a few seconds of CPU time, no command takes more than a tenth of a second of CPU time.

B.9. Program SELEC3.FOR

This program is used to select subsets of measurements using the mean of all interclass distances as selection criterion. It is basically identical to the program SELEC2.FOR, except that it uses subsets of

inverse matrices in the computations. In consequence, it requires as input one file per class, where the mean vector and inverse covariance matrix are stored (as created by the program INVERT.FOR). For more details, the listing of the program should be consulted.

APPENDIX C

PROGRAM LISTINGS

```
MAIN.   PATGEN.FOR      FORTRAN V.S(515) /KI    26-JUL-77       17:30   PAGE 1


00001   C
00002   C       THIS PROGRAM WILL READ FROM A DISK FILE A NUMBER OF
00003   C       IMAGES AS CHARACTERIZED BY A LIST OF COORDINATE PAIRS
00004   C       AND GENERATE 7 OTHER SETS BY SYMMETRIZATION AROUND
00005   C       VARIOUS AXES. THEN SCATTER POINTS WILL BE GENERATED
00006   C       AROUND EACH COORDINATE PAIR (=REFLEXION POINT);
00007   C       NSC POINTS WILL FORM THE SCATTER, IN A CIRCULAR NORMAL
00008   C       DISTRIBUTION AROUND THE REFLEXION POINT ( HOWEVER,
00009   C       SOME POINTS MAY BE DELETED IF THEY OCCUR AT THE SAME
00010   C       LOCATION AS PREVIOUSLY GENERATED POINTS.
00011   C
00012           PROGRAM PATGEN
00013           PARAMETER NIMAG=12,NCD=30,NMAX=70
00014   C
00015   C       NIMAG IS THE MAXIMUM NUMBER OF INPUT IMAGES
00016   C       NCD IS THE MAXIMUM NUMBER OF INPUT COORDINATE PAIRS
00017   C       NMAX IS THE MAXIMUM NUMBER OF SCATTER POINTS
00018   C
00019           INTEGER IM(1:2000,1:8,1:2)
00020   C
00021   C       IM CONTAINS THE COORDINATE PAIRS
00022   C
00023           DOUBLE PRECISION FIN,FOUT
00024   C
00025           TYPE 100                        ASK FOR INPUT FILENAME
00026           ACCEPT 200,FIN
00027   1       TYPE 101                        ASK FOR NUMBER OF IMAGES
00028           ACCEPT 201,NIM
00029           IF (NIM .LE. 0 .OR. NIM .GT. NIMAG) GO TO 1
00030   2       TYPE 103                        ASK FOR NUMBER OF SC. PTS
00031           ACCEPT 201,NSC
00032           IF (NSC .LE. 0 .OR. NSC .GT. NMAX) GO TO 2
00033           TYPE 102                        ASK FOR OUTPUT FILENAME
00034           ACCEPT 200,FOUT
00035           OPEN (MODE='ASCII',UNIT=20,DEVICE='DSK',FILE=FIN)
00036           OPEN (MODE='ASCII',UNIT=21,DEVICE='DSK',FILE=FOUT)
00037   C
00038   C       BOTH INPUT AND OUTPUT FILES ARE NOW OPEN
00039   C
00040           IX = 0
00041           IY = 0                          FOR RANDOM NUMBER GENERATOR
00042           DO 90 I=1,NIM                   GENERAL LOOP FOR IMAGES
00043           J = 0
00044   5       J = J+1                         NEXT COORDINATE PAIR
00045           IF (J .GT. NCD) STOP 'E R R O R'
00046           READ (20,202) (IM(J,1,K),K=1,2)  READ IN A PAIR
00047   C
00048   C       NOW WE CREATE 7 OTHER PAIRS
00049   C
00050           IM(J,2,1) = -IM(J,1,1)
00051           IM(J,2,2) = +IM(J,1,2)          SYMM. AROUND Y-AXIS
00052           IM(J,3,1) = +IM(J,1,1)
00053           IM(J,3,2) = -IM(J,1,2)          SYMM. AROUND X-AXIS
00054           IM(J,4,1) = +IM(J,1,2)
00055           IM(J,4,2) = +IM(J,1,1)          SYMM. AROUND 1ST BISECTOR
00056           IM(J,5,1) = +IM(J,1,2)
```

```
PATGEN  PATGEN.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-1


00057           IM(J,5,2) = +IM(J,1,1)
00058           IM(J,6,1) = +IM(J,1,1)
00059           IM(J,6,2) = -IM(J,1,2)
00060           IM(J,7,1) = -IM(J,1,1)
00061           IM(J,7,2) = +IM(J,1,2)
00062           IM(J,8,1) = +IM(J,1,1)
00063           IM(J,8,2) = +IM(J,1,2)
00064           IF (IM(J,1,1) .NE. 256 .AND. IM(J,1,2) .NE. 256) GO TO 5
00065   C
00066   C       256,256 IS THE TERMINATOR PAIR
00067   C
00068           NCP = J-1                       NUMBER OF COORDINATE PAIRS READ
00069           DO 90 II=1,8                    LOOP ON THE EIGHT SETS
00070           IND = NCP                       POINTS TO THE LAST PAIR WRITTEN
00071           DO 80 J=1,NCP                   LOOP FOR SCATTER AROUND EACH PT
00072           NX = IM(J,II,1)
00073           NY = IM(J,II,2)
00074           DO 80 K=1,NSC                   GENERATE NSC SCATTER POINTS
00075           CALL SCAT(IX,IY,NX,NY,NNX,NNY)    NEW PAIR IN NNX,NNY
00076           DO 60 L=1,IND                   CHECK FOR EXIST. OF SAME PAIR
00077           IF (NNX .EQ. IM(L,II,1) .AND. NNY .EQ. IM(L,II,2)) GO TO 80
00078   C
00079   C       IF THE PAIR ALREADY EXISTS, IT IS IGNORED
00080   C
00081   60      CONTINUE
00082   C
00083   C       THE NEW PAIR IS VALID, SO WRITE IT
00084   C
00085           IND = IND+1
00086           IM(IND,II,1) = NNX
00087           IM(IND,II,2) = NNY
00088   80      CONTINUE
00089   C
00090   C       A SEPARATOR PAIR WILL NOW BE ADDED
00091   C
00092           IND = IND+1
00093           IM(IND,II,1) = 256
00094           IM(IND,II,2) = 256
00095   C
00096   C       THE IMAGE IS NOW COMPLETE AND CAN BE WRITTEN TO DISK
00097   C
00098           WRITE (21,203) ((IM(K,II,L),L=1,2),K=1,IND)
00099   90      CONTINUE                        NEXT IMAGE
00100           CLOSE (UNIT=20,DEVICE='DSK',FILE=FIN)
00101           CLOSE (UNIT=21,DEVICE='DSK',FILE=FOUT)
00102           STOP
00103   C
00104   C       END OF PROGRAM
00105   C
00106   C       FORMAT BLOCK
00107   C
00108   100     FORMAT (///,T10,'TYPE IN INPUT FILENAME : ',$)
00109   101     FORMAT (T10,'TYPE IN NUMBER OF IMAGES IN FILE : ',$)
00110   102     FORMAT (T10,'TYPE IN OUTPUT FILENAME : ',$)
00111   103     FORMAT (T10,'TYPE IN NUMBER OF SCATTER POINTS : ',$)
00112   C
```

```
PATGEN  PATGEN.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-2


00113   200     FORMAT (A10)
00114   201     FORMAT (I2)
00115   202     FORMAT (2I)
00116   203     FORMAT (I4,',',I4)
00117   C
00118           END


SUBPROGRAMS CALLED

SCAT


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NY     1       *K      2       *NX     3       *NCP    4       *NIM    5
 FOUT   6        FIN     10      *J      12       .S0007 13       .S0006 14
 .S0005 15       .S0004 16       *NNX    17       .S0003 20      *II     21
 .S0002 22       .S0001 23        .S0000 24      *IND    25      *IY     26
*L      27      *I      30      *NSC    31      *NNY    32       IM     33
*IX     76433

TEMPORARIES


PATGEN    NO ERRORS DETECTED
```

```
MAIN.   PATGEN.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE SCAT(IX,IY,NX,NY,NNX,NNY)
00005   C
00006   C       NTIM IS THE NUMBER OF RANDOM VAR. SUMMED TO GET A NORMAL
00007   C       DISTRIBUTION
00008   C
00009           INTEGER IX,IY,NX,NY,NNX,NNY
00010           SD = 2.5
00011           SLIM = 3.*SD
00012   10      CALL GAUSS(IX,IY,SD,DUM)
00013   C
00014   C       DUM IS NORMAL, WITH ZERO MEAN AND STAND. DEV. OF SD
00015   C
00016           IF (ABS(DUM) .GT. SLIM) GO TO 10
00017           NNX = NX+IFIX(DUM)
00018   20      CALL GAUSS(IX,IY,SD,DUM)
00019           IF (ABS(DUM) .GT. SLIM) GO TO 20         KEEP WITHIN 3 ST. DEV.
00020           NNY = NY+IFIX(DUM)
00021           RETURN
00022   C
00023   C       END OF ROUTINE
00024   C
00025           END


SUBPROGRAMS CALLED

GAUSS
IFIX.   ABS.

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 NY     1       *SD     2       *SLIM   3       NX      4       NNX     5
*DUM    6        IY     7        NNY    10      IX      11

TEMPORARIES

 .A0016 12

SCAT      NO ERRORS DETECTED
```

```
MAIN.   PATGEN.FOR      FORTRAN V.S(515) /KI    26-JUL-77       17:30   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE GAUSS(IX,IY,SD,DUM)
00005           REAL SD,DUM
00006           INTEGER IX,IY
00007           DUM = 0.
00008           DO 10 IDUM=1,12
00009           DUM = DUM+RAN(IX,IY)-0.5
00010   10      CONTINUE
00011           DUM = SD*DUM
00012           RETURN
00013           END

SUBPROGRAMS CALLED

RAN

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 SD     1       *IDUM   2       .S0000 3        DUM    4        IY     5
 IX     6

TEMPORARIES

 .A0016 7

GAUSS     NO ERRORS DETECTED
```

```
MAIN.   MOMENT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1


00001   C
00002   C       THIS PROGRAM COMPUTES CENTRAL MOMENTS OF CENTRAL
00003   C       IMAGES PERTAINING TO THE SAME CLASS
00004   C       ALL MOMENTS OF ORDER 2 TO NRD - NRD IS USER SPECIFIED -
00005   C       ARE COMPUTED AND SCALED BY THE ZERO-TH ORDER MOMENT
00006   C       THE MOMENT VECTORS REPRESENTATIVE OF EACH IMAGE ARE
00007   C       THEN WRITTEN IN A SPECIFIED FILE
00008   C       THE IMAGES ARE FOUND IN A SPECIFIED FILE ON DISK
00009   C       AN IMAGE IS A LIST OF COORDINATE PAIRS, FINISHING
00010   C       BY THE PAIR (256,256)
00011   C
00012           PROGRAM MOMENT
00013           PARAMETER NIMAG=100,NORD=12,NMAX=91,NCLASS=20
00014   C
00015   C       NIMAG IS THE MAXIMUM NUMBER OF IMAGES THAT CAN BE
00016   C       PROCESSED AT ONE TIME
00017   C       NORD IS THE MAXIMUM ORDER OF MOMENT THAT CAN BE COMPUTED
00018   C       (THE NUMBER OF MOMENTS OF ORDER <= NORD IS GIVEN BY
00019   C       NMAX = (NORD+1)*(NORD+2)/2 )
00020   C
00021           REAL MOM(1:NMAX,1:NIMAG)
00022           INTEGER IM(1:1000,1:2)
00023           DOUBLE PRECISION FIN(1:NCLASS),FOUT(1:NCLASS)
00024   C
00025   C       MOM WILL CONTAIN THE MOMENTS OF ALL PATTERNS OF
00026   C       A CLASS. IM WILL BE USED AS TEMPORARY STORAGE FOR
00027   C       AN IMAGE. FIN AND FOUT CONTAIN THE INPUT AND OUTPUT
00028   C       FILENAMES FOR ALL CLASSES
00029   C
00030   1       TYPE 100                        ASK FOR NUMBER OF CLASSES
00031           ACCEPT 200,NCL
00032           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00033   2       TYPE 101                        ASK FOR NUMBER OF IMAGES
00034           ACCEPT 200,NIM
00035           IF (NIM .LE. 0 .OR. NIM .GT. NIMAG) GO TO 2
00036   3       TYPE 102                        ASK FOR MAX ORDER OF MOMENTS
00037           ACCEPT 200,NRD
00038           IF (NRD .LT. 0 .OR. NRD .GT. NORD) GO TO 3
00039           TYPE 103                        ASK FOR CONSTANT IN INDEX FUNCTION
00040           ACCEPT 203,CST                  ANY REAL VALUE IS ACCEPTABLE
00041           NTOT = (NRD+1)*(NRD+2)/2-3
00042           CST = CST*512.
00043   C
00044   C       NUMERICAL INPUT CHECKED
00045   C
00046           DO 10 I=1,NCL
00047           TYPE 104,I                      ASK FOR INPUT FILENAME
00048           ACCEPT 201,FIN(I)
00049           TYPE 105                        ASK FOR OUTPUT FILENAME
00050           ACCEPT 201,FOUT(I)
00051   10      CONTINUE
00052   C
00053   C       DIALOGUE FINISHED
00054   C
00055           DO 90 I=1,NCL                   GENERAL LOOP FOR CLASSES
00056           OPEN (MODE='ASCII',UNIT=21,DEVICE='DSK',FILE=FIN(I))
```

```
MOMENT  MOMENT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-1


00057   C
00058   C       THE INPUT FILE FOR THE ITH CLASS IS NOW OPEN
00059   C
00060           DO 85 J=1,NIM                   LOOP FOR IMAGES IN A CLASS
00061           K = 0
00062   15      K = K+1                         NEXT COORDINATE PAIR
00063           IF (K .GT. 1000) GO TO 95               ERROR
00064           READ (21,202) (IM(K,L),L=1,2)   READ A COORDINATE PAIR
00065           IF (IM(K,1) .NE. 256 .AND. IM(K,2) .NE. 256) GO TO 15
00066   C
00067   C       256,256 IS THE TERMINATOR PAIR
00068   C
00069           NCP = K-1                       NUMBER OF COORDINATE PAIRS
00070   C
00071   C       THE PROGRAM WILL NOW COMPUTE THE CENTRAL MOMENTS OF THE
00072   C       IMAGE STORED IN IM.
00073   C       THE INDEX FUNCTION IS DUE TO A. BARRERO
00074   C
00075           XAV = 0.
00076           YAV = 0.
00077           DO 20 K=1,NCP
00078           XTMP = IM(K,1)+255
00079           YTMP = IM(K,2)+255
00080           XAV = XAV+(1.+XTMP/CST)/FLOAT(NCP)
00081           YAV = YAV+(1.+YTMP/CST)/FLOAT(NCP)
00082   20      CONTINUE
00083   C
00084   C       THIS LOOP HAS COMPUTED THE COORDINATES OF THE CENTER OF GRAVITY
00085   C
00086           NN = 0
00087           DO 80 K=2,NRO
00088           DO 80 N1=0,K
00089           NN = NN+1
00090           N2 = K-N1
00091   C
00092   C       N1 AND N2 ARE THE TWO EXPONENTS FOR THE INDICES
00093   C
00094           MOM(NN,J) = 0.
00095           DO 80 M=1,NCP
00096           X = IM(M,1)+255
00097           Y = IM(M,2)+255
00098           X = 1.+X/CST-XAV
00099           Y = 1.+Y/CST-YAV
00100           MOM(NN,J) = MOM(NN,J)+(X**N1)*(Y**N2)/FLOAT(NCP)
00101   80      CONTINUE
00102   C
00103   C       ALL MOMENTS OF AN IMAGE  HAVE NOW BEEN COMPUTED
00104   C       SO PROCESS NEXT IMAGE
00105   C
00106   85      CONTINUE
00107   C
00108   C       ALL MOMENTS OF A CLASS HAVE NOW BEEN COMPUTED
00109   C       THEY WILL BE WRITTEN TO FILE AND THE NEXT CLASS
00110   C       WILL BE PROCESSED
00111   C
00112           CLOSE (UNIT=21,DEVICE='DSK',FILE=FIN(I))
```

```
MOMENT  MOMENT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-2


00113           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=FOUT(I))
00114           WRITE (20,203) ((MOM(K,L),K=1,NTOT),L=1,NIM)
00115           CLOSE (UNIT=20,DEVICE='DSK',FILE=FOUT(I))
00116   90      CONTINUE
00117   C
00118   C       ALL CLASSES PROCESSED : END OF PROGRAM
00119   C
00120           STOP
00121   C
00122   95      TYPE 150
00123           STOP                            ERROR
00124   C
00125   C       FORMAT BLOCK
00126   C
00127   100     FORMAT (///,T10,'THIS PROGRAM WILL COMPUTE THE CENTRAL
00128           1 MOMENTS',/,T10,'OF A IMAGE. THIS WILL BE DONE
00129           2 FOR EACH SPECIFIED CLASS.',/,T10,'UP TO THE SPECIFIED
00130           3 NUMBER OF IMAGES PER CLASS.',/,T10,'PLEASE ANSWER
00131           4 THE FOLLOWING QUESTIONS, FINISHING BY CR.',///,T10,
00132           5'TYPE IN NUMBER OF CLASSES : ',$)
00133   101     FORMAT (T10,'TYPE IN NUMBER OF IMAGES PER CLASS : ',$)
00134   102     FORMAT (T10,'TYPE IN MAXIMUM ORDER OF MOMENTS : ',$)
00135   103     FORMAT (T10,'TYPE IN THE INDEX CONSTANT (REAL #) : ',$)
00136   104     FORMAT (T10,'TYPE IN INPUT FILENAME FOR CLASS',I3,' : ',$)
00137   105     FORMAT (T10,'TYPE IN OUTPUT FILENAME FOR THIS CLASS : ',$)
00138   C
00139   150     FORMAT (///,T20,'E R R O R  ',///)
00140   C
00141   200     FORMAT (I3)
00142   201     FORMAT (A10)
00143   202     FORMAT (2I)
00144   203     FORMAT (E14.7)
00145   C
00146   C       END OF FORMAT BLOCK
00147   C       AND END OF PROGRAM
00148   C
00149           END


SUBPROGRAMS CALLED

FLOAT.


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "$" NOT REFERENCED

*NTOT   1       *N1     2       *K      3       *NCL    4       *NCP    5
*NRD    6       *CST    7       *Y      10      *YTMP   11      *NIM    12
 FOUT   13       FIN    63      *M      133     *J      134      .S0007 135
 .S0006 136      .S0005 137       .S0004 140     *YAV    141      .S0003 142
 .S0002 143      .S0001 144       .S0000 145      MOM    146     *X      21762
*XTMP   21763    .S0011 21764     .S0010 21765   *L      21766   *I      21767
 IM     21770   *XAV    25710    *N2     25711   *NN     25712
```

```
MOMENT  MOMENT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-3

TEMPORARIES

 .Q0000 26102    .Q0001 26103

MOMENT    NO ERRORS DETECTED
```

```
MAIN.   MOMINV.FOR      FORTRAN V.S(515) /KI    27-JUL-77       11:57   PAGE 1


00001   C
00002   C       THIS PROGRAM COMPUTES THE SIX MOMENT INVARIANTS OF SECOND
00003   C       AND THIRD ORDER FROM A SERIES OF PICTURES FOUND IN SPECI-
00004   C       FIED FILES, AND WRITES THEM TO A SPECIFIED FILE.
00005   C       THE PICTURES SHOULD ALL PERTAIN TO THE SAME CLASS.
00006   C       THE FIRST INVARIANT (GYRATION RADIUS) IS USED TO SCALE
00007   C       THE NEXT SIX, AND IS NOT WRITTEN TO DISK.
00008   C       THE LAST INVARIANT IS WRITTEN IN ABSOLUTE VALUE, AS
00009   C       IT CHANGES SIGN UNDER IMPROPER TRANSFORMATION.
00010   C       AN IMAGE IS AN ARRAY OF NUMBERS STORED ON DISK IN
00011   C       PACKED (4 BYTES PER WORD) FORM. THE ROUTINE UNPACK
00012   C       (WRITTEN IN ASSEMBLER) IS USED TO UNPACK THE WORDS
00013   C
00014           PROGRAM MOMINV
00015           PARAMETER NDIM=128,NDM=32,NPIC=16,NMAX=100
00016   C
00017   C       NDIM IS THE MAXIMUM SIZE OF A PICTURE
00018   C       NDM = NDIM/4
00019   C       NPIC IS THE MAXIMUM NUMBER OF PICTURES PER CLASS (ARBITRARY)
00020   C       NMAX=NPIC*6 IS THE TOTAL NUMBER OF PICTURES PROCESSED
00021   C       (AS 5 NEW PICTURES ARE GENERATED BY ROTATION AND SYMMETRY
00022   C       FROM EACH ORIGINAL PICTURE)
00023   C
00024           COMMON PIC(1:NDIM,1:NDIM),PIC2(1:NDIM,1:NDIM)
00025           COMMON INV(1:6,1:NMAX)
00026           INTEGER VEC(1:NDIM),INP(1:NDM)
00027           INTEGER PIC,PIC2
00028           REAL INV
00029           DOUBLE PRECISION D(1:NPIC),DOUT
00030   C
00031   C       PIC CONTAINS THE PICTURE TO BE PROCESSED
00032   C       PIC2 IS TEMPORARY STORAGE USED FOR  ROTATING PICTURES
00033   C       INP AND VEC ARE USED IN CONJUNCTION WITH UNPACK
00034   C       INV CONTAINS THE INVARIANTS AS COLUMN VECTORS
00035   C       D CONTAINS THE INPUT (PICTURE) FILENAMES
00036   C       DOUT CONTAINS THE OUTPUT (INVARIANTS) FILENAME
00037   C
00038   1       TYPE 100                        ASK FOR # OF PICTURES
00039           ACCEPT 200,NPC
00040           IF (NPC .LE. 0 .OR. NPC .GT. NPIC) GO TO 1
00041   2       TYPE 101                        ASK FOR SIZE OF PICTURES
00042           ACCEPT 200,NSZ
00043           IF (NSZ .LE. 0 .OR. NSZ .GT. NDIM) GO TO 2
00044           DO 10 I=1,NPC
00045           TYPE 102,I                      ASK FOR INPUT FILENAME
00046           ACCEPT 201,D(I)
00047   10      CONTINUE
00048           TYPE 103                        ASK FOR OUTPUT FILENAME
00049           ACCEPT 201,DOUT
00050   C
00051   C       INITIALIZATION FINISHED
00052   C
00053           NTOT = 6*NPC
00054           NZ = NSZ/4
00055           DO 70 K=1,NPC                   GENERAL LOOP FOR PICTURES
00056           OPEN (MODE='IMAGE',UNIT=20,DEVICE='DSK',FILE=D(K))
```

```
MOMINV  MOMINV.FOR      FORTRAN V.S(515) /KI    27-JUL-77       11:57   PAGE 1-1


00057   C
00058   C       THE INPUT FILENAME IS OPEN
00059   C       DATA WILL BE READ COLUMN BY COLUMN, AND IMMEDIATELY
00060   C       UNPACKED
00061   C
00062           DO 25 J=1,NSZ                           LOOP THROUGH THE COLUMNS
00063           READ (20) (INP(I),I=1,NZ)
00064           CALL UNPACK(NZ,INP,VEC)                 UNPACKS INP INTO VEC
00065           DO 25 I=1,NSZ
00066   25      PIC2(I,J) = VEC(I)
00067   C
00068   C       THE ENTIRE PICTURE HAS NOW BEEN READ
00069   C       AND UNPACKED
00070   C
00071           CLOSE (UNIT=20,DEVICE='DSK',FILE=D(K))
00072   C
00073           KK = 6*(K-1)+1
00074   C
00075   C       THE FIRST PICTURE USED IS THE ORIGINAL
00076   C
00077           DO 30 I=1,NSZ
00078           DO 30 J=1,NSZ
00079   30      PIC(I,J) = PIC2(I,J)
00080           CALL COMP(NSZ,KK)                       COMPUTE INVARIANTS
00081   C
00082   C       THE SECOND PICTURE USED IS THE ORIGINAL ROTATED 90 DEGREES
00083   C       CLOCKWISE
00084   C
00085           KK = KK+1
00086           DO 40 I=1,NSZ
00087           II = NSZ+1-I                            SYMMETRIZE AROUND Y-AXIS
00088           DO 40 J=1,NSZ
00089
00090   40      PIC(I,J) = PIC2(J,II)
00091           CALL COMP(NSZ,KK)
00092   C
00093   C       THE THIRD PICTURE USED IS THE ORIGINAL ROTATED 90 DEGREES
00094   C       ANTI CLOCKWISE
00095   C
00096           KK = KK+1
00097           DO 50 I=1,NSZ
00098           DO 50 J=1,NSZ
00099           JJ = NSZ+1-J                            SYMMETRIZE AROUND X-AXIS
00100   50      PIC(I,J) = PIC2(JJ,I)
00101           CALL COMP(NSZ,KK)                       COMPUTE INVARIANTS
00102   C
00103   C       THE FOURTH PICTURE USED IS THE ORIGINAL ROTATED 180 DEGREES
00104   C
00105           KK = KK+1
00106           DO 60 I=1,NSZ
00107           II = NSZ+1-I
00108           DO 60 J=1,NSZ
00109           JJ = NSZ+1-J
00110   60      PIC(I,J) = PIC2(II,JJ)
00111           CALL COMP(NSZ,KK)
00112   C
```

```
00113   C       THE FIFTH PICTURE USED IS THE ORIGINAL SYMMETRIZED AROUND
00114   C       THE Y-AXIS
00115   C
00116           KK = KK+1
00117           DO 63 I=1,NSZ
00118           II = NSZ+1-I
00119           DO 63 J=1,NSZ
00120   63      PIC(I,J) = PIC2(II,J)
00121           CALL COMP(NSZ,KK)
00122   C
00123   C       THE SIXTH AND LAST PICTURE USED IS THE ORIGINAL
00124   C       SYMMETRIZED AROUND THE X-AXIS
00125   C
00126           KK = KK+1
00127           DO 66 I=1,NSZ
00128           DO 66 J=1,NSZ
00129           JJ = NSZ+1-J
00130   66      PIC(I,J) = PIC2(I,JJ)
00131           CALL COMP(NSZ,KK)
00132   C
00133   C       THIS ORIGINAL PICTURE HAS NOW BEEN ENTIRELY PROCESSED
00134   C       THE NEXT PICTURE WILL BE USED
00135   C
00136   70      CONTINUE
00137   C
00138   C       ALL PICTURES HAVE BEEN PROCESSED
00139   C       THE INVARIANTS WILL BE WRITTEN TO DISK
00140   C
00141           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=DOUT)
00142           WRITE (20,202) ((INV(I,J),I=1,6),J=1,NTOT)
00143           CLOSE (UNIT=20,DEVICE='DSK',FILE=DOUT)
00144   C
00145   C       NOW THE INVARIANTS WILL BE PRINTED
00146   C       IN 6 COLUMNS OF 6 NUMBERS
00147   C
00148           DO 90 K=1,NPC
00149           KK = 6*(K-1)+1
00150           PRINT 204                                   SEPARATOR
00151           PRINT 203,((INV(I,J),J=KK,KK+5),I=1,6)
00152   90      CONTINUE
00153   C
00154   C       END OF ROUTINE
00155   C
00156           STOP
00157   C
00158   C
00159   C       FORMAT BLOCK
00160   C
00161   100     FORMAT (///,T10,'THIS PROGRAM COMPUTES 6 MOMENT INVARIANTS
00162           1 FROM',/,T10,'SPECIFIED PICTURES OF A CLASS.',//,T10,
00163           2'ENTER NUMBER OF PICTURES IN CLASS : ',$)
00164   101     FORMAT (T10,'ENTER DIMENSION OF PICTURES :  ',$)
00165   102     FORMAT (T10,'ENTER INPUT FILENAME FOR PICTURE',I3,' : ',$)
00166   103     FORMAT (T10,'ENTER OUTPUT FILENAME : ',$)
00167   C
00168   200     FORMAT (I3)
```

```
MOMINV  MOMINV.FOR      FORTRAN V.5(515) /KI    27-JUL-77       11:57   PAGE 1-3


00169    201      FORMAT (A10)
00170    202      FORMAT (E14.7)
00171    203      FORMAT (6('       ',E14.7))
00172    204      FORMAT (///)
00173    C
00174    C        END OF FORMAT BLOCK
00175    C        AND END OF PROGRAM
00176    C
00177             END


COMMON BLOCKS

/.COMM./(+101130)
PIC     +0       PIC2    +40000  INV     +100000


SUBPROGRAMS CALLED

COMP    UNPACK

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 .S0020 I        *JJ     2        INP     3        DOUT    43       *NTOT   45
*K      46       *NSZ    47      *NPC     50      *J      51        .S0007 52
 .S0006 53       *KK     54       .S0005 55       D       56        .S0004 116
*II     117       .S0003 120       .S0002 121      VEC     122       .S0001 322
 .S0000 323       .S0017 324       .S0016 325       .S0015 326       .S0014 327
 .S0013 330       .S0012 331       .S0011 332       .S0010 333      *I      334
*NZ     335       .S0025 336       .S0024 337       .S0023 340       .S0022 341
 .S0021 342

TEMPORARIES

 .Q0000 444       .Q0001 445

MOMINV    NO ERRORS DETECTED
```

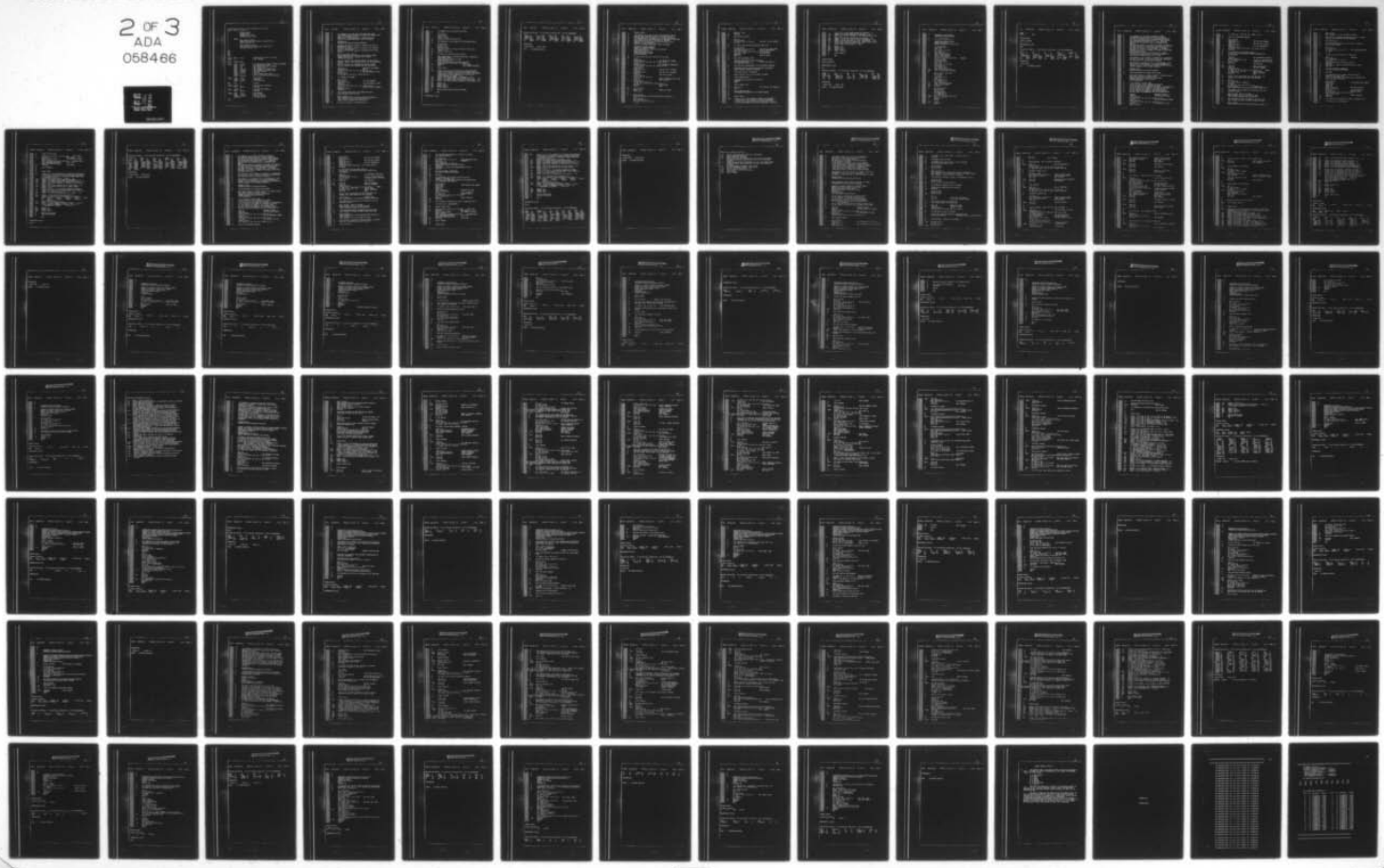
```
MAIN.   MOMINV.FOR      FORTRAN V.5(515) /KI    27-JUL-77       11:57   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE COMP(NSZ,KK)
00005           PARAMETER NDIM=128,NDM=32,NPIC=16,NMAX=100
00006   C
00007   C       THIS ROUTINE COMPUTES THE SIX MOMENT INVARIANTS AND
00008   C       WRITES THEM TO THE ARRAY INV
00009   C
00010           COMMON PIC(1:NDIM,1:NDIM),PIC2(1:NDIM,1:NDIM)
00011           COMMON INV(1:6,1:NMAX)
00012           INTEGER PIC,PIC2,KK,NSZ
00013           REAL INV
00014           VM00 = 0.
00015           VM01 = 0.
00016           VM10 = 0.                       INITIALIZE NON CENTRAL MOMENTS OF ORDER
00017           DO 10 I=1,NSZ
00018           DO 10 J=1,NSZ
00019           VM00 = VM00+PIC(I,J)
00020           VM01 = VM01+FLOAT(J)*PIC(I,J)
00021   10      VM10 = VM10+FLOAT(I)*PIC(I,J)
00022           XAV = VM10/VM00
00023           YAV = VM01/VM00                 TO FIND THE CENTER OF GRAVITY
00024   C
00025   C       NOW THE SEVEN MOMENTS OF ORDER 2 AND 3 WILL BE COMPUTED
00026   C
00027           VM20 = 0.
00028           VM11 = 0.
00029           VM02 = 0.
00030           VM30 = 0.
00031           VM21 = 0.
00032           VM12 = 0.
00033           VM03 = 0.                       INITIALIZE
00034           DO 20 I=1,NSZ
00035           FIN = FLOAT(I)-XAV              CENTRAL INDEX
00036           DO 20 J=1,NSZ
00037           FJN = FLOAT(J)-YAV              CENTRAL INDEX
00038           VM20 = VM20+FIN*FIN*PIC(I,J)/(VM00*VM00)
00039           VM11 = VM11+FIN*FJN*PIC(I,J)/(VM00*VM00)
00040           VM02 = VM02+FJN*FJN*PIC(I,J)/(VM00*VM00)
00041           VM30 = VM30+FIN*FIN*FIN*PIC(I,J)/(VM00**2.5)
00042           VM21 = VM21+FIN*FIN*FJN*PIC(I,J)/(VM00**2.5)
00043           VM12 = VM12+FIN*FJN*FJN*PIC(I,J)/(VM00**2.5)
00044   20      VM03 = VM03+FJN*FJN*FJN*PIC(I,J)/(VM00**2.5)
00045           GYR = VM20+VM02                 GYRATION RADIUS
00046   C
00047   C       NOW COMPUTE THE SIX INVARIANTS
00048   C
00049           INV(1,KK) = 4.*(VM11**2)+((VM20-VM02)**2)
00050           INV(1,KK) = INV(1,KK)/(GYR**2)
00051           INV(2,KK) = ((VM30-3.*VM12)**2)+((3.*VM21-VM03)**2)
00052           INV(2,KK) = INV(2,KK)/(GYR**3)
00053           INV(3,KK) = ((VM12+VM30)**2)+((VM21+VM03)**2)
00054           INV(3,KK) = INV(3,KK)/(GYR**3)
00055           INV(4,KK) = (VM30-3.*VM12)*(VM30+VM12)*(((VM30+VM12)**2)
00056          1-3.*((VM21+VM03)**2)) + (3.*VM21-VM03)*(VM21+VM03)*
```

```
COMP    MOMINV.FOR .     FORTRAN V.5(515) /KI    27-JUL-77       11:57   PAGE 1-1


00057            2(3.*((VM30+VM12)**2)-((VM21+VM03)**2))
00058            INV(4,KK) = INV(4,KK)/(GYR**6)
00059            INV(5,KK) = 4.*VM11*(VM30+VM12)*(VM21+VM03) + (VM20-VM02)*
00060            1(((VM30+VM12)**2)-((VM21+VM03)**2))
00061            INV(5,KK) = INV(5,KK)/(GYR**6)
00062            INV(6,KK) = (3.*VM21-VM03)*(VM30+VM12)*(((VM30+VM12)**2)
00063            1-3.*((VM21+VM03)**2)) - (VM30-3.*VM12)*(VM21+VM03)*
00064            2(3.*((VM30+VM12)**2)-((VM21+VM03)**2))
00065            INV(6,KK) = ABS(INV(6,KK))/(GYR**6)
00066    C
00067    C       ALL INVARIANTS HAVE NOW BEEN COMPUTED
00068    C       THE ROUTINE IS FINISHED
00069    C
00070            RETURN
00071            END


COMMON BLOCKS

/.COMM./(+101130)
PIC     +0      PIC2    +40000  INV     +100000


SUBPROGRAMS CALLED

FLOAT.  ABS.

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*VM03   1       *VM20   2       *VM02   3       *FJN    4       NSZ     5
*VM30   6       *VM01   7       *GYR    10      *FIN    11      *VM12   12
*VM00   13      *J      14       KK     15      *YAV    16       .S0003 17
*VM11   20       .S0002 21       .S0001 22       .S0000 23      *VM10   24
*I      25      *VM21   26      *XAV    27

TEMPORARIES

 .A0016 30

COMP      NO ERRORS DETECTED
```

AD-A058 466 TENNESSEE UNIV KNOXVILLE DEPT OF ELECTRICAL ENGINEERING F/G 9/2
A RECURSIVE PROCEDURE FOR UPDATING QUADRATIC FORMS AND ITS APPL--ETC(U)
JAN 78 R C GONZALEZ, B M MORET N00014-75-C-0545

UNCLASSIFIED ONR-CR215-288-3F NL

2 OF 3
ADA
058466



```
TITLE PACKER. ROUTINE TO PACK AND UNPACK 9 BIT BYTES
SUBTTL <TH> MAY 17, 1977
;
;               THOMAS HANDLER
;               PHYSICS DEPT.
;               UNIVERSITY OF TENN.
;               KNOXVILLE TENN. 37916
;
;       USAGE:
;               CALL PACK(N,IN,IOUT)
;               PACKS 4*N 9 BIT BYTES FROM 4*N WORDS OF IOUT
;               INTO N WORDS OF IN
;
;               CALL UNPACK(N,IN,IOUT)
;               UNPACKS 4*N 9 BIT BYTES FROM N WORDS OF IN
;               INTO 4*N WORDS OF IOUT
;
A=0
B=1
C=2
D=3
P=17
;
PIXEL:  POINT   9,(B)           ; BYTE POINTER TO GET 9 BIT BYTE
PIX1:   Z                       ; TEMP STORAGE
;
ENTRY   UNPACK, PACK
;
SETUP:  MOVN    A,@(16)         ; GET NEGATIVE OF NUMBER OF WORDS TO PROCESS
        JUMPE   A,RET           ; IF ZERO NOTHING TO DO
        IMULI   A,4             ; GET NUMBER OF BYTES TO PROCESS
        HRLM    A,C             ; SET UP FOR LOOP
        MOVEI   B,@1(16)        ; GET ADDRESS OF PACKEED ARRAY
        HRRI    C,@2(16)        ; GET ADDRESS OF UNPACKED ARRAY
        MOVE    A,PIXEL         ; GET BYTE POINTER
        MOVEM   A,PIX1          ; SAVE IT
        SETZM   A               ; ZERO INTERMEDIATE STORAGE
        JUMPE   D,LOOP2         ; IF AC D NOT ZERO WE ARE UNPACKING
;
LOOP1:  ILDB    A,PIX1          ; GET ELEMENT
        MOVEM   A,(C)           ; STORE IT
        AOBJN   C,LOOP1         ; ARE THERE MORE
;
RET:    POPJ    P,              ; RETURN
;
UNPACK: SETOM   D               ; SET FLAG FOR UNPACKING
        JRST    SETUP           ; PROCEED
;
PACK:   SETZM   D               ; SET FLAG FOR PACKING
        JRST    SETUP           ; PROCEED
;
LOOP2:  MOVE    A,(C)           ; GET ELEMENT
        IDPB    A,PIX1          ; PACK IT AWAY
        AOBJN   C,LOOP2         ; SEE IF ANY MORE
        JRST RET                ; NO MORE. RETURN
;
END
```

```
MAIN.   STAT.FOR        FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1


00001   C
00002   C       THIS PROGRAM WILL READ FROM SPECIFIED DISK FILES
00003   C       THE PATTERN VECTORS OF SEVERAL CLASSES AND COMPUTE THEIR
00004   C       MEAN AND COVARIANCE MATRIX
00005   C       THESE WILL IN TURN BE WRITTEN TO SPECIFIED FILES
00006   C       INPUT IS IN SINGLE PRECISION, OUTPUT IN DOUBLE
00007   C
00008           PROGRAM STAT
00009           PARAMETER NCLASS=20,NIMAG=100,NCOMP=91
00010   C
00011   C       NCLASS IS THE MAXIMUM NUMBER OF CLASSES THAT CAN BE
00012   C       PROCESSED AT ONCE
00013   C       NIMAG IS THE MAXIMUM NUMBER OF IMAGES IN THE TRAINING
00014   C       SET FOR A GIVEN CLASS
00015   C       NCOMP IS THE MAXIMUM NUMBER OF COMPONENTS PER PATTERN
00016   C       VECTOR
00017   C
00018           REAL VEC(1:NCOMP,1:NIMAG)
00019           DOUBLE PRECISION MEAN(1:NCOMP),COV(1:NCOMP,1:NCOMP)
00020           DOUBLE PRECISION FN1(1:NCLASS),FN2(1:NCLASS)
00021           INTEGER NI(1:NCLASS)
00022   C
00023   C       VEC WILL CONTAIN THE PATTERN VECTORS - AS COL. VECT.-
00024   C       MEAN WILL CONTAIN THE MEAN VECTOR AND COV THE COVARIANCE
00025   C       MATRIX
00026   C       FN1 WILL CONTAIN THE FILENAMES FOR THE DATA INPUT
00027   C       FN2 WILL CONTAIN THE FILENAMES FOR THE DATA OUTPUT
00028   C
00029   1       TYPE 100                                ASK FOR # OF CLASSES.
00030           ACCEPT 200,NCL                          GET IT
00031           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00032   2       TYPE 101                                ASK FOR NUMBER OF COMP.
00033           ACCEPT 200,NC
00034           IF (NC .LE. 0 .OR. NC .GT. NCOMP) GO TO 2
00035   C
00036   C       ALL ANSWERS ARE CORRECT
00037   C
00038           DO 5 I=1,NCL
00039           TYPE 102,I                              ASK FOR FILENAME
00040           ACCEPT 201,FN1(I)
00041   4       TYPE 103                                ASK FOR NUMBER OF IMAG.
00042           ACCEPT 200,NI(I)
00043           IF (NI(I) .LE. 1 .OR. NI(I) .GT. NIMAG) GO TO 4
00044           TYPE 104                                ASK FOR OUTPUT FILENAME
00045           ACCEPT 201,FN2(I)
00046   5       CONTINUE
00047   C
00048   C       THE FILENAMES ARE KNOWN; THE PROGRAM WILL NOW
00049   C       DEAL WITH EACH CLASS IN TURN
00050   C
00051           DO 30 L=1,NCL
00052           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=FN1(L))
00053           READ (20,202) ((VEC(I,J),I=1,NC),J=1,NI(L))
00054           CLOSE (UNIT=20,DEVICE='DSK',FILE=FN1(L))
00055   C
00056   C       THE DATA FOR CLASS L HAVE BEEN READ IN
```

```
STAT    STAT.FOR        FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-1


00057   C       THE PROGRAM WILL NOW COMPUTE THE MEAN
00058   C
00059           TN = NI(L)
00060           DO 10 I=1,NC
00061           MEAN(I) = 0.
00062           DO 10 J=1,NI(L)
00063   10      MEAN(I) = MEAN(I)+VEC(I,J)/TN
00064   C
00065   C       THE PROGRAM WILL NOW COMPUTE THE COVARIANCE MATRIX
00066   C
00067           DO 25 I=1,NC
00068           DO 25 J=I,NC
00069           COV(I,J) = 0.
00070           DO 20 K=1,NI(L)
00071   20      COV(I,J) = COV(I,J)+(VEC(I,K)-MEAN(I))*(VEC(J,K)-
00072           1MEAN(J))/TN
00073   25      COV(J,I) = COV(I,J)
00074   C
00075   C.      THE DATA WILL NOW BE WRITTEN TO DISK
00076   C
00077           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=FN2(L))
00078   C       WRITE MEAN VECTOR
00079           WRITE (20,203) (MEAN(I),I=1,NC)
00080   C       WRITE COV. MATRIX
00081           WRITE (20,203) ((COV(I,J),J=1,NC),I=1,NC)
00082           CLOSE (UNIT=20,DEVICE='DSK',FILE=FN2(L))
00083   30      CONTINUE                                NEXT CLASS
00084           STOP                                    ALL CLASSES PROCESSED
00085   C
00086   C       FORMAT BLOCK
00087   C
00088   100     FORMAT (///,T10,'THIS PROGRAM WILL COMPUTE THE MEAN
00089           1 AND THE COVARIANCE',/,T10,'MATRIX OF SPECIFIED CLAS
00090           2SES.',/,T10,'SEVERAL PARAMETERS ARE NEEDED. ANSWER THE
00091           3 QUESTIONS.',/,T10,'FINISHING BY A CR.',//,T10,
00092           4'ENTER NUMBER OF CLASSES : ',$)
00093   101     FORMAT (T10,'ENTER NUMBER OF COMPONENTS : ',$)
00094   102     FORMAT (T10,'ENTER INPUT FILENAME FOR CLASS',I3,' : ',$)
00095   103     FORMAT (T10,'ENTER NUMBER OF IMAGES IN THIS CLASS : ',$)
00096   104     FORMAT (T10,'ENTER OUTPUT FILENAME FOR THIS CLASS : ',$)
00097   C
00098   200     FORMAT (I3)
00099   201     FORMAT (A10)
00100   202     FORMAT (E14.7)
00101   203     FORMAT (D22.15)
00102   C
00103   C       END OF FORMAT BLOCK AND END OF PROGRAM
00104   C
00105           END

SUBPROGRAMS CALLED
```

```
STAT    STAT.FOR        FORTRAN V.5(515) /KI    26-JUL-77       17:30   PAGE 1-2


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 MEAN    1       *K      267     *NCL    270     FN1     271     *J      341
 .S0007 342       .S0006 343      .S0005 344      .S0004 345      .S0003 346
 .S0002 347       VEC    350      .S0001 22164    COV    22165    .S0000 62447
 FN2    62450     .S0013 62520    .S0012 62521    .S0011 62522    .S0010 62523
*L      62524    *I      62525    NI     62526   *TN     62552   *NC     62553


TEMPORARIES

 .Q0000 62706    .Q0001 62707

STAT      NO ERRORS DETECTED
```

```
MAIN.   INVERT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:31   PAGE 1


00001           PROGRAM INVERT
00002   C
00003   C       THIS PROGRAM WILL READ ON DISK THE COVARIANCE MATRICES
00004   C       COMPUTE THEIR INVERSES AND WRITE THEM BACK ON DISK
00005   C       THE USER SPECIFIES THE NAMES OF ALL FILES, THE NUMBER
00006   C       OF MATRICES, THEIR DIMENSION, AND THE COMPONENTS TO BE USED
00007   C       EACH COVARIANCE MATRIX IS PRECEDED BY THE MEAN
00008   C       VECTOR OF THE CLASS, WHICH IS ALSO WRITTEN IN FIRST
00009   C       POSITION IN THE INVERSE FILE
00010   C       INPUT IS IN DOUBLE PRECISION, OUTPUT IN SINGLE
00011   C
00012           PARAMETER NCLASS=50,NCOMP=91
00013           COMMON C1(1:NCOMP,1:NCOMP)
00014           DOUBLE PRECISION D1(1:NCLASS),D2(1:NCLASS)
00015           REAL VMEAN(1:NCOMP)
00016           DOUBLE PRECISION C1,DVAL
00017           INTEGER SING,IND(1:NCOMP)
00018           DATA IND/NCOMP*0/
00019   C
00020   C       D1 AND D2 WILL CONTAIN THE NAMES OF THE FILES
00021   C
00022   1       TYPE 100
00023           ACCEPT 200,NCL                          GET NUMBER OF CLASSES
00024           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00025   2       TYPE 101
00026           ACCEPT 200,NSZ                          GET NUMBER OF COMPONENTS
00027           IF (NSZ .LE. 0 .OR. NSZ .GT. NCOMP) GO TO 2
00028           DO 10 I=1,NCL
00029           TYPE 102,I
00030           ACCEPT 201,D1(I)                        GET COV. MAT. FILENAME
00031           TYPE 103
00032           ACCEPT 201,D2(I)                        GET INV. MAT. FILENAME
00033   10      CONTINUE
00034           TYPE 105                                ASK FOR ALL COMP. ?
00035           ACCEPT 200,NYN
00036           IF (NYN .EQ. 1) GO TO 12
00037           TYPE 106                                COMP. INTRODUCED ONE BY ONE
00038   11      TYPE 107                                ASK FOR COMP #
00039           ACCEPT 200,NB
00040           IF (NB .EQ. 0) GO TO 14
00041           IF (NB .LE. 0 .OR. NB .GT. NSZ) GO TO 11
00042           IND(NB) = 1                             SELECT COMP.
00043           GO TO 11                                NEXT
00044   C
00045   12      DO 13 I=1,NSZ
00046   13      IND(I) = 1                              SELECT ALL COMP.
00047           NSZA = NSZ
00048   C
00049   C
00050   14      DO 20 K=1,NCL
00051           OPEN(MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D1(K))
00052           I = 0
00053           DO 16 J=1,NSZ
00054           READ (20,203),DVAL
00055           IF (IND(J) .EQ. 0) GO TO 16
00056           I = I+1
```

```
INVERT  INVERT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:31   PAGE 1-1

00057           VMEAN(I) = DVAL
00058   16      CONTINUE
00059   C
00060   C       READ THE MEAN VECTOR
00061   C
00062           II = 0
00063           DO 15 I=1,NSZ
00064           IF (IND(I) .NE. 0) II=II+1          THIS COL. WILL BE STORED
00065           JJ = 0
00066   C
00067   C       II  AND JJ ARE THE ROW AND COLUMN INDEX IN C1
00068   C
00069           DO 15 J=1,NSZ
00070           IF (IND(J) .NE. 0) JJ=JJ+1          THIS ROW WILL BE STORED
00071           READ (20,203),DVAL                  READ IN ONE ELEMENT
00072           IF (IND(I) .EQ. 0 .OR. IND(J) .EQ. 0) GO TO 15
00073           C1(JJ,II) = DVAL                    STORE ELEMENT
00074   15      CONTINUE
00075           NSZA = II                           SIZE OF MATRIX
00076   C
00077   C       READ THE COVARIANCE MATRIX
00078   C
00079           CLOSE(UNIT=20,DEVICE='DSK',FILE=D1(K))
00080           CALL INV(NSZA,SING)                 INVERT THE MATRIX C1
00081           IF (SING .EQ. 1) GO TO 50
00082   C
00083   C       THE MATRIX WAS NON-SINGULAR, SO THE PROGRAM PROCEEDS
00084   C
00085           OPEN(MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D2(K))
00086           WRITE (20,202) (VMEAN(I),I=1,NSZA)
00087   C
00088   C       WRITE MEAN VECTOR ( UNCHANGED )
00089   C
00090           WRITE (20,202) ((C1(I,J),I=1,NSZA),J=1,NSZA)
00091   C
00092   C       WRITE INVERSE MATRIX
00093   C
00094           CLOSE(UNIT=20,DEVICE='DSK',FILE=D2(K))
00095   20      CONTINUE
00096           STOP
00097   C
00098   C       END OF NORMAL PATH
00099   C
00100   50      TYPE 104,K                          THE K-TH MAT. IS SINGULAR
00101           STOP
00102   C
00103   C       END OF SINGULAR PATH
00104   C       THE PROGRAM IS ABORTED WITH AN ERROR MESSAGE
00105   C
00106   C
00107   C       FORMAT BLOCK
00108   C
00109   100     FORMAT (///,T5,'THIS PROGRAM COMPUTES THE INVERSES OF
00110           1 COVARIANCE',/,T5,'MATRICES AS FOUND IN SPECIFIED
00111           2 FILES.',/,T5,'AND WRITES THEM TO SPECIFIED FILES.',
00112           3/,T5,'PLEASE ANSWER THE FOLLOWING QUESTIONS, FINISHING
```

```
INVERT  INVERT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:31   PAGE 1-4

00113           4 BY A CR.',//,T10,'ENTER NUMBER OF MATRICES : ',$)
00114   101     FORMAT (T10,'ENTER DIMENSION OF MATRICES : ',$)
00115   102     FORMAT (T10,'ENTER FILENAME OF COVARIANCE MATRIX #',I3,
00116           1' : ',$)
00117   103     FORMAT (T10,'ENTER FILENAME FOR ITS INVERSE : ',$)
00118   104     FORMAT (//,T10,'THE',I3,'TH MATRIX IS SINGULAR  ',//)
00119   105     FORMAT (T10,'DO YOU WANT ALL COMP. (YES=1/NO=0) : ',$)
00120   106     FORMAT (T10,'TYPE IN DESIRED COMP. INDICES ONE BY ONE.',
00121           1/,T10,'WHEN FINISHED, TYPE A ZERO.',/)
00122   107     FORMAT (T10,'TYPE IN COMP. # : ',$)
00123   C
00124   200     FORMAT (I3)
00125   201     FORMAT (A10)
00126   202     FORMAT (E14.7)
00127   203     FORMAT (D22.15)
00128   C
00129   C       END OF PROGRAM
00130   C
00131           END

.

COMMON BLOCKS

/.COMM./(+40262)
C1      +0

SUBPROGRAMS CALLED

INV

SCALARS AND ARRAYS   "+" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 SING    1       *JJ     2       *N8     3       *K      4       *NCL    5
*NSZA    6       *NSZ    7        D2     10       DVAL   154     *J      156
 .S0007 157       .S0006 160       D1     161      .S0005 325      .S0004 326
*II     327       .S0003 330       .S0002 331      .S0001 332      .S0000 333
 IND    334       .S0010 467      *NYN    470     *I      471      VMEAN  472

TEMPORARIES

 .Q0000 1027     .Q0001 1030

INVERT    NO ERRORS DETECTED
```

```
MAIN.   INVERT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:31   PAGE 1


00001   C
00002           SUBROUTINE INV(NSZ,SING)
00003           PARAMETER NCOMP=91,NK=4200
00004   C
00005   C       NK = NCOMP*(NCOMP+1)/2
00006   C
00007           COMMON C1(1:NCOMP,1:NCOMP)
00008           DOUBLE PRECISION C1
00009           DOUBLE PRECISION A(1:NK),R(1:NK)
00010   C
00011           DO 100 J=1,NSZ
00012           JJ = NSZ*(J-1)+J
00013           TEMP = 0.
00014           IF (J .EQ. 1) GO TO 6
00015           DO 5 M=1,J-1
00016           JM = NSZ*(J-1)+M
00017   5       TEMP = TEMP+R(JM)**2
00018   6       TEMP2 = C1(J,J)-TEMP
00019           R(JJ) = SQRT(TEMP2)
00020           IF (R(JJ) .EQ. 0.) GO TO 150      SINGULAR
00021           A(JJ) = 1./R(JJ)
00022           DO 55 I=J+1,NSZ
00023           IJ = NSZ*(I-1)+J
00024           TEMP = 0.
00025           IF (J .EQ. 1) GO TO 55
00026           DO 50 M=1,J-1
00027           IM = NSZ*(I-1)+M
00028           JM = NSZ*(J-1)+M
00029   50      TEMP = TEMP+R(JM)*R(IM)
00030   55      R(IJ) = A(JJ)*(C1(I,J)-TEMP)
00031   C
00032           IF (J .EQ. 1) GO TO 100
00033           DO 80 L=2,J
00034           K = J-L+1
00035           JK = NSZ*(J-1)+K
00036           TEMP = 0.
00037           DO 75 M=K,J-1
00038           JM = NSZ*(J-1)+M
00039           MK = NSZ*(M-1)+K
00040   75      TEMP = TEMP+R(JM)*A(MK)
00041   80      A(JK) = -A(JJ)*TEMP
00042   100     CONTINUE
00043   C
00044           DO 110 J=1,NSZ
00045           DO 110 I=1,J
00046           C1(J,I) = 0.
00047           DO 105 K=J,NSZ
00048           KI = NSZ*(K-1)+I
00049           KJ = NSZ*(K-1)+J
00050   105     C1(J,I) = C1(J,I)+A(KI)*A(KJ)
00051   110     C1(I,J) = C1(J,I)
00052           SING = 0
00053           RETURN
00054   C
00055   150     SING = 1
00056           RETURN
```

```
INV     INVERT.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:31   PAGE 1-1

00057   C
00058           END


COMMON BLOCKS

/.COMM./(+40262)
C1      +0

SUBPROGRAMS CALLED

SQRT.

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*SING   1       *JJ     2       *K      3       *NSZ    4       *TEMP2  5
*IJ     6       *M      7       *J      10       .S0007 11      *JM     12
 .S0006 13       .S0005 14       .S0004 15       .S0003 16       A      17
 .S0002 20337    .S0001 20340    .S0000 20341   *KJ     20342   *TEMP   20343
 R      20344    .S0010 40664   *L      40665    .I0000 40666   *I      40667
*MK     40670   *IM     40671   *JK     40672   *KI     40673

TEMPORARIES

 .A0016 40674

INV       NO ERRORS DETECTED
```

```
MAIN.   CLASS1.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:42   PAGE 1


00001   C
00002   C       THIS PROGRAM WILL CLASSIFY DIFFERENT PATTERNS
00003   C       BY USING THE MINIMUM-DISTANCE DECISION CRITERION
00004   C       THE DISTANCE MEASURE USED IS THE MAHALANOBIS DISTANCE
00005   C       WITH EACH CLASS BEING CHARACTERIZED BY A MEAN VECTOR
00006   C       AND AN INVERSE COVARIANCE MATRIX ( OR A PRINCIPAL
00007   C       SUBMATRIX OF THE LATTER IF NOT ALL COMPONENTS ARE
00008   C       USED ).
00009   C       THE USER MUST SUPPLY THE NUMBER OF CLASSES, THE NAMES OF
00010   C       THE FILES WHERE THE PATTERNS AND THE STATISTICS ARE TO
00011   C       BE FOUND, AND THE #S OF THE COMPONENTS TO BE USED
00012   C       EACH SET OF PATTERNS ( ONE PER CLASS ) IS STORED IN A
00013   C       DIFFERENT FILE, AND THE NUMBER OF PATTERNS IN THE FILES
00014   C       MUST BE SUPPLIED TOO.
00015   C       THE TRAINING PATTERNS WILL USUALLY BE OF  LARGER
00016   C       DIMENSION THAN THE INVERSE COVARIANCE MATRICES, BUT
00017   C       THE SELECTED COMPONENTS ONLY WILL BE USED
00018   C
00019   C       THIS PROGRAM IS NOT INTENDED TO PERFORM AS A CLASSIFIER,
00020   C       BUT ONLY TO BE USED IN THE VERIFICATION OF EXPERIMENTAL
00021   C       DATA OBTAINED WITH THE SELECTION PROGRAMS.
00022   C
00023   C       THE OUTPUT IS A EXTENSIVE LIST OF THE DISTANCES ( FROM
00024   C       EACH PATTERN TO EACH CLASS ) AND THE CLASSIFICATION
00025   C       NUMBER. A LIST OF CORRECT AND INCORRECT CLASSIFICATIONS
00026   C       IS ALSO PRINTED, WITH A PERCENT ESTIMATE.
00027   C
00028           PROGRAM CLASS1
00029           PARAMETER NCLASS=10,NIMAG=100,NCOMP=91
00030   C
00031   C       THESE PARAMETERS ONLY DEFINE UPPER BOUNDS
00032   C
00033           REAL MEAN(1:NCOMP),C(1:NCOMP,1:NCOMP),V(1:NCOMP,1:NIMAG)
00034           REAL DIST(1:NCLASS,1:NCLASS,1:NIMAG)
00035           DOUBLE PRECISION D1(1:NCLASS),D2(1:NCLASS)
00036           INTEGER IND(1:NCOMP),NIM(1:NCLASS)
00037           DATA IND/NCOMP*0/
00038   C
00039   C       M WILL CONTAIN A CLASS' MEAN
00040   C       C WILL CONTAIN A CLASS' INVERSE COV. MATRIX
00041   C       DIST WILL CONTAIN ALL COMPUTED DISTANCES
00042   C       V WILL CONTAIN THE PATTERNS OF THE CLASS IN CONSIDERATION
00043   C       NIM WILL CONTAIN THE NUMBER OF PATTERNS IN EACH SET
00044   C       D1 WILL CONTAIN THE NAMES OF THE PATTERN FILES
00045   C       D2 WILL CONTAIN THE NAMES OF THE STATISTICS FILES
00046   C       IND WILL FLAG THE COMPONENTS USED FOR CLASSIFICATION
00047   C
00048           TYPE 99                                 MESSAGE TO USER
00049   1       TYPE 100                                ASK FOR CLASS NUMBER
00050           ACCEPT 200,NCL
00051           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00052   2       TYPE 101                                ASK FOR COMPONENT NUMBER
00053           ACCEPT 200,NCO
00054           IF (NCO .LE. 0 .OR. NCO .GT. NCOMP) GO TO 2
00055   9       TYPE 109                                ASK FOR DIM. OF INV.
00056           ACCEPT 200,NDM
```

```
CLASS1  CLASS1.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:42   PAGE 1-1

00057           IF  (NDM .LE. 0 .OR. NDM .GT. NCOMP) GO TO 9
00058   C
00059   C       FIRST TWO PARAMETERS CHECKED
00060   C
00061           DO 15 I=1,NCL
00062           TYPE 103,I                          ASK FOR 1ST FILENAME
00063           ACCEPT 201,D1(I)
00064           TYPE 104                            ASK FOR 2ND FILENAME
00065           ACCEPT 201,D2(I)
00066   14      TYPE 102                            ASK FOR COMPONENTS #
00067           ACCEPT 200,NIM(I)
00068           IF (NIM(I) .LE. 0 .OR. NIM(I) .GT. NIMAG) GO TO 14
00069   15      CONTINUE
00070   C
00071   C       ALL FILENAMES HAVE NOW BEEN TYPED IN
00072   C       LAST OF ALL, THE SELECTED COMPONENTS NUMBERS WILL
00073   C       BE REQUESTED
00074   C
00075           TYPE 105                            ALL COMPONENTS SELECTED ?
00076           ACCEPT 200,NYN
00077           IF (NYN .EQ. 0) GO TO 25            IF NOT, ASK FOR SELECTION
00078           DO 20 I=1,NCO                       OTHERWISE, SELECT THEM ALL
00079   20      IND(I) = I
00080           GO TO 35                            AND CONTINUE
00081   C
00082   25      TYPE 106                            MODE OF SELECTION
00083   30      TYPE 107                            ASK FOR A NUMBER
00084           ACCEPT 200,NMB
00085           IF (NMB .LT. 0 .OR. NMB .GT. NCO) GO TO 30      ERROR
00086           IF (NMB .EQ. 0) GO TO 35            END OF INPUT
00087           IND(NMB) = NMB                      SELECT COMPONENT
00088           GO TO 30                            NEXT
00089   C
00090   C       NOW ALL INPUT PARAMETERS HAVE BEEN OBTAINED, THE
00091   C       PROGRAM WILL START COMPUTING THE DISTANCES
00092   C
00093           NMB = 0
00094   35      DO 40 K=1,NCO
00095           IF (IND(K) .NE. 0) NMB=NMB+1
00096   40      CONTINUE
00097           IF (NMB .NE. NDM) STOP 'E R R O R'
00098           DO 80 K=1,NCL                       LARGEST LOOP
00099           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D2(K))
00100   C
00101   C       THE PROGRAM WILL NOW READ IN THE MEAN AND INV. COV.
00102   C       MATRIX OF CLASS K
00103   C
00104           READ (20,202) (MEAN(I),I=1,NDM)
00105           READ (20,202) ((C(I,J),J=1,NDM),I=1,NDM)
00106           CLOSE (UNIT=20,DEVICE='DSK',FILE=D2(K))
00107   C
00108   C       NOW THE PROGRAM ENTERS A SECONDARY LOOP THAT WILL
00109   C       LOOK THROUGH THE CLASS PATTERNS, CLASS PER CLASS
00110   C
00111           DO 80 L=1,NCL
00112           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D1(L))
```

```
CLASS1  CLASS1.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:42   PAGE 1-2

00113           NIMM = NIM(L)
00114           READ (20,202) ((V(I,J),I=1,NCO),J=1,NIMM)
00115   C
00116   C       NOW THE SET OF TRAINING PATTERNS OF CLASS L HAS BEEN
00117   C       READ IN. THE DISTANCES WILL BE COMPUTED
00118   C
00119           DO 80 M=1,NIM(L)
00120   C
00121   C       THE DISTANCE FROM PATTERN M OF CLASS L TO CLASS K
00122   C       WILL NOW BE COMPUTED
00123   C
00124           II = 0
00125           DO 75 N=1,NCO
00126           IF (IND(N) .EQ. 0) GO TO 75       SKIP UNSELECTED COMP.
00127           II = II+1
00128           JJ = 0
00129   C
00130   C       II AND JJ ARE THE INDICES IN C AND MEAN
00131   C
00132           PROD = 0.                         TEMPORARY
00133           DO 70 NN=1,NCO
00134           IF (IND(NN) .EQ. 0) GO TO 70
00135           JJ = JJ+1
00136           PROD = PROD+C(II,JJ)*(MEAN(JJ)-V(NN,M))
00137   70      CONTINUE
00138           DIST(K,L,M) = DIST(K,L,M)+PROD*(MEAN(II)-V(N,M))
00139   75      CONTINUE
00140   C
00141   C       NOW THE DISTANCE IS COMPUTED
00142   C       SO GO ON TO COMPUTE THE OTHERS
00143   C
00144   80      CONTINUE
00145   C
00146   C       ALL DISTANCES HAVE NOW BEEN COMPUTED AND ARE
00147   C       STORED IN THE ARRAY DIST
00148   C       THEY WILL BE PRINTED AND EXAMINED FOR CLASSIFICATION
00149   C       ON A CLASS PER CLASS BASIS
00150   C
00151           DO 90 L=1,NCL
00152           TYPE 150,L                        TYPE SOURCE CLASS NUMBER
00153           PRINT 150,L
00154           PRINT 153
00155           NCORR = 0
00156           DO 87 M=1,NIM(L)                  EXAMINE EACH COMP.
00157           VMIN = DIST(1,L,M)
00158           IMIN = 1                          INIT. CLASSIFIER
00159           DO 85 K=1,NCL                     CLASSIF. LOOP
00160           IF (VMIN .LE. DIST(K,L,M)) GO TO 85
00161           VMIN = DIST(K,L,M)
00162           IMIN = K                          FOUND A SMALLER
00163   85      CONTINUE
00164   C
00165   C       THE PATTERN IS CLASSIFIED IN CLASS K (SMALLEST DIST.)
00166   C
00167           IF (IMIN .EQ. L) NCORR=NCORR+1
00168   C
```

```
CLASS1  CLASS1.FOR      FORTRAN V.S(515) /KI    26-JUL-77       17:42   PAGE 1-3


00169   C       ADJUST CLASS. COUNT
00170   C
00171           PRINT 151,IMIN,(DIST(I,L,M),I=1,NCL)     PRINT INFO
00172   87      CONTINUE                                 NEXT COMP.
00173           NFALS = NIM(L)-NCORR                     GET NUMBER OF ERRORS
00174           NIMM = NIM(L)
00175           PER = 100.*FLOAT(NCORR)/FLOAT(NIMM)      SUCCESS PERC.
00176           TYPE 152,NCORR,NFALS,PER                 TYPE INFO
00177           PRINT 152,NCORR,NFALS,PER
00178   90      CONTINUE                                 NEXT CLASS
00179   C
00180   C       END OF PROGRAM
00181   C
00182   C
00183   C       FORMAT BLOCK
00184   C
00185   99      FORMAT (//,T10,'THIS PROGRAM WILL CLASSIFY YOUR TRAINING
00186           1 PATTERNS',/,T10,'USING A MINIMUM DISTANCE CLASSIFIER',
00187           2/,T10,'SOME INFORMATION IS REQUIRED; ANSWER QUESTIONS,
00188           3 FINISHING BY CR',//)
00189   100     FORMAT (T10,'ENTER NUMBER OF CLASSES : ',$)
00190   101     FORMAT (T10,'ENTER TOTAL NUMBER OF COMPONENTS : ',$)
00191   102     FORMAT (T10,'ENTER NUMBER OF PATTERNS IN THIS SET : ',$)
00192   103     FORMAT (T10,'ENTER FILENAME OF TRAINING SET OF CLASS',
00193           1I3,' : ',$)
00194   104     FORMAT (T10,'ENTER FILENAME OF INV. COV. MATRIX : ',$)
00195   105     FORMAT (T10,'DO YOU WANT TO USE ALL COMP. (YES=1,
00196           1NO=0) ? ',$)
00197   106     FORMAT (T10,'TYPE IN THE DESIRED COMPONENTS NUMBERS,
00198           1 ONE AT A TIME.',/,T10,'WHEN FINISHED, TYPE A ZERO.',/)
00199   107     FORMAT (T10,'TYPE COMP. # : ',$)
00200   109     FORMAT (T10,'TYPE IN DIMENSION OF INV. MATRIX :  ',$)
00201   C
00202   150     FORMAT (///,T30,'RESULTS FOR CLASS',I3,//)
00203   153     FORMAT (T2,'CLASSIF.:',20X,'DISTANCES TO CLASSES:',/,
00204           2T10,' CLASS 1     CLASS 2     CLASS 3     CLASS 4     CLASS
00205           3 5     CLASS 6     CLASS 7     CLASS 8     CLASS 9
00206           4     CLASS 10',//)
00207   151     FORMAT (T3,I3,2X,10(E10.3,2X))
00208   152     FORMAT (/,T10,'--NUMBER OF CORRECT CLASSIF.   :',I3,/,
00209           1T10,'  NUMBER OF ERRONEOUS CLASSIF. :',I3,/,T10,
00210           2'  PERCENTAGE OF SUCCESS      :',F10.5,/)
00211   C
00212   200     FORMAT (I3)
00213   201     FORMAT (A10)
00214   202     FORMAT (E14.7)
00215   C
00216   C       END OF FORMAT BLOCK
00217   C       AND END OF PROGRAM
00218   C
00219           END


SUBPROGRAMS CALLED

FLOAT.
```

```
CLASS1  CLASS1.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:42   PAGE 1-4


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 .S0020 1       *NCO    2       *JJ     3        MEAN    4       *N      137
*K      140     *NCL    141     *PROD   142     *NFALS  143     *NOM    144
*IMIN   145      D2     146      NIM    172      V      204     *M      22020
*J      22021    .S0007 22022    .S0006 22023    D1     22024    .S0005 22050
*NIMM   22051    .S0004 22052   *II     22053    .S0003 22054    .S0002 22055
 DIST   22056    .S0001 45476    .S0000 45477   *NMB    45500    .S0017 45501
 IND    45502    .S0016 45635    .S0015 45636    .S0014 45637    .S0013 45640
 .S0012 45641    .S0011 45642    .S0010 45643   *PER    45644   *L      45645
*NYN    45646   *I      45647   *VMIN   45650    C      45651   *NCORR  66002
*NN     66003

TEMPORARIES

 .Q0000 66326    .Q0001 66327

CLASS1    NO ERRORS DETECTED
```

```
MAIN.   CLASS2.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:43   PAGE 1


00001   C
00002   C       THIS PROGRAM WILL CLASSIFY DIFFERENT PATTERNS
00003   C       BY USING THE MINIMUM-DISTANCE DECISION CRITERION
00004   C       THE DISTANCE MEASURE USED IS THE MAHALANOBIS DISTANCE
00005   C       WITH EACH CLASS BEING CHARACTERIZED BY A MEAN VECTOR
00006   C       AND AN INVERSE COVARIANCE MATRIX ( OR A PRINCIPAL
00007   C       SUBMATRIX OF THE LATTER IF NOT ALL COMPONENTS ARE
00008   C       USED ).
00009   C       THE USER MUST SUPPLY THE NUMBER OF CLASSES, THE NAMES OF
00010   C       THE FILES WHERE THE PATTERNS AND THE STATISTICS ARE TO
00011   C       BE FOUND, AND THE #S OF THE COMPONENTS TO BE USED
00012   C       EACH SET OF PATTERNS ( ONE PER CLASS ) IS STORED IN A
00013   C       DIFFERENT FILE, AND THE NUMBER OF PATTERNS IN THE FILES
00014   C       MUST BE SUPPLIED TOO.
00015   C
00016   C       THIS PROGRAM IS NOT INTENDED TO PERFORM AS A CLASSIFIER,
00017   C       BUT ONLY TO BE USED IN THE VERIFICATION OF EXPERIMENTAL
00018   C       DATA OBTAINED WITH THE SELECTION PROGRAMS.
00019   C
00020   C       THE OUTPUT IS A EXTENSIVE LIST OF THE DISTANCES ( FROM
00021   C       EACH PATTERN TO EACH CLASS ) AND THE CLASSIFICATION
00022   C       NUMBER. A LIST OF CORRECT AND INCORRECT CLASSIFICATIONS
00023   C       IS ALSO PRINTED, WITH A PERCENT ESTIMATE.
00024   C
00025           PROGRAM CLASS2
00026           PARAMETER NCLASS=10,NIMAG=64,NCOMP=91
00027   C
00028   C       THESE PARAMETERS ONLY DEFINE UPPER BOUNDS
00029   C
00030           REAL MEAN(1:NCOMP),C(1:NCOMP,1:NCOMP),V(1:NCOMP,1:NIMAG)
00031           REAL DIST(1:NCLASS,1:NCLASS,1:NIMAG)
00032           DOUBLE PRECISION D1(1:NCLASS),D2(1:NCLASS)
00033           INTEGER IND(1:NCOMP),NIM(1:NCLASS)
00034           DATA IND/NCOMP*0/
00035   C
00036   C       M WILL CONTAIN A CLASS' MEAN
00037   C       C WILL CONTAIN A CLASS' INVERSE COV. MATRIX
00038   C       DIST WILL CONTAIN ALL COMPUTED DISTANCES
00039   C       V WILL CONTAIN THE PATTERNS OF THE CLASS IN CONSIDERATION
00040   C       NIM WILL CONTAIN THE NUMBER OF PATTERNS IN EACH SET
00041   C       D1 WILL CONTAIN THE NAMES OF THE PATTERN FILES
00042   C       D2 WILL CONTAIN THE NAMES OF THE STATISTICS FILES
00043   C       IND WILL FLAG THE COMPONENTS USED FOR CLASSIFICATION
00044   C
00045           TYPE 99                             MESSAGE TO USER
00046   1       TYPE 100                            ASK FOR CLASS NUMBER
00047           ACCEPT 200,NCL
00048           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00049   2       TYPE 101                            ASK FOR COMPONENT NUMBER
00050           ACCEPT 200,NCO
00051           IF (NCO .LE. 0 .OR. NCO .GT. NCOMP) GO TO 2
00052   9       TYPE 109                            ASK FOR DIM. OF INV.
00053           ACCEPT 200,NDM
00054           IF  (NDM .LE. 0 .OR. NDM .GT. NCOMP) GO TO 9
00055   C
00056   C       FIRST TWO PARAMETERS CHECKED
```

```
CLASS2  CLASS2.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:43   PAGE 1-1


00057   C
00058           DO 15 I=1,NCL
00059           TYPE 103,I                              ASK FOR 1ST FILENAME
00060           ACCEPT 201,D1(I)
00061           TYPE 104                                ASK FOR 2ND FILENAME
00062           ACCEPT 201,D2(I)
00063   14      TYPE 102                                ASK FOR COMPONENTS #
00064           ACCEPT 200,NIM(I)
00065           IF (NIM(I) .LE. 0 .OR. NIM(I) .GT. NIMAG) GO.TO 14
00066   15      CONTINUE
00067   C
00068   C       ALL FILENAMES HAVE NOW BEEN TYPED IN
00069   C       LAST OF ALL, THE SELECTED COMPONENTS NUMBERS WILL
00070   C       BE REQUESTED
00071   C
00072           TYPE 105                                ALL COMPONENTS SELECTED ?
00073           ACCEPT 200,NYN
00074           IF (NYN .EQ. 0) GO TO 25                IF NOT, ASK FOR SELECTION
00075           DO 20 I=1,NCO                           OTHERWISE, SELECT THEM ALL
00076   20      IND(I) = I
00077           GO TO 35                                AND CONTINUE
00078   C
00079   25      TYPE 106                                MODE OF SELECTION
00080   30      TYPE 107                                ASK FOR A NUMBER
00081           ACCEPT 200,NMB
00082           IF (NMB .LT. 0 .OR. NMB .GT. NCO) GO TO 30      ERROR
00083           IF (NMB .EQ. 0) GO TO 35                END OF INPUT
00084           IND(NMB) = NMB                          SELECT COMPONENT
00085           GO TO 30                                NEXT
00086   C
00087   C       NOW ALL INPUT PARAMETERS HAVE BEEN OBTAINED, THE
00088   C       PROGRAM WILL START COMPUTING THE DISTANCES
00089   C
00090   35      DO 80 K=1,NCL                           LARGEST LOOP
00091           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D2(K))
00092   C
00093   C       THE PROGRAM WILL NOW READ IN THE MEAN AND INV. COV.
00094   C       MATRIX OF CLASS K
00095   C
00096           READ (20,202) (MEAN(I),I=1,NDM)
00097           READ (20,202) ((C(I,J),J=1,NDM),I=1,NDM)
00098           CLOSE (UNIT=20,DEVICE='DSK',FILE=D2(K))
00099   C
00100   C       NOW THE PROGRAM ENTERS A SECONDARY LOOP THAT WILL
00101   C       LOOK THROUGH THE CLASS PATTERNS, CLASS PER CLASS
00102   C
00103           DO 80 L=1,NCL
00104           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D1(L))
00105           NIMM = NIM(L)
00106           READ (20,202) ((V(I,J),I=1,NCO),J=1,NIMM)
00107   C
00108   C       NOW THE SET OF TRAINING PATTERNS OF CLASS L HAS BEEN
00109   C       READ IN. THE DISTANCES WILL BE COMPUTED
00110   C
00111           DO 80 M=1,NIM(L)
00112   C
```

```
CLASS2  CLASS2.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:43   PAGE 1-2


00113   C       THE DISTANCE FROM PATTERN M OF CLASS L TO CLASS K
00114   C       WILL NOW BE COMPUTED
00115   C
00116           DO 75 N=1,NCO
00117           IF (IND(N) .EQ. 0) GO TO 75             SKIP UNSELECTED COMP.
00118           PROD = 0.                               TEMPORARY
00119           DO 70 NN=1,NCO
00120           IF (IND(NN) .EQ. 0) GO TO 70
00121           PROD = PROD+C(N,NN)*(MEAN(NN)-V(NN,M))
00122   70      CONTINUE
00123           DIST(K,L,M) = DIST(K,L,M)+PROD*(MEAN(N)-V(N,M))
00124   75      CONTINUE
00125   C
00126   C       NOW THE DISTANCE IS COMPUTED
00127   C       SO GO ON TO COMPUTE THE OTHERS
00128   C
00129   80      CONTINUE
00130   C
00131   C       ALL DISTANCES HAVE NOW BEEN COMPUTED AND ARE
00132   C       STORED IN THE ARRAY DIST
00133   C       THEY WILL BE PRINTED AND EXAMINED FOR CLASSIFICATION
00134   C       ON A CLASS PER CLASS BASIS
00135   C
00136           DO 90 L=1,NCL
00137           TYPE 150,L                              TYPE SOURCE CLASS NUMBER
00138           PRINT 150,L
00139           PRINT 153
00140           NCORR = 0
00141           DO 87 M=1,NIM(L)                        EXAMINE EACH COMP.
00142           VMIN = DIST(1,L,M)
00143           IMIN = 1                                INIT. CLASSIFIER
00144           DO 85 K=1,NCL                           CLASSIF. LOOP
00145           IF (VMIN .LE. DIST(K,L,M)) GO TO 85
00146           VMIN = DIST(K,L,M)
00147           IMIN = K                                FOUND A SMALLER
00148   85      CONTINUE
00149   C
00150   C       THE PATTERN IS CLASSIFIED IN CLASS K (SMALLEST DIST.)
00151   C
00152           IF (IMIN .EQ. L) NCORR=NCORR+1
00153   C
00154   C       ADJUST CLASS. COUNT
00155   C
00156           PRINT 151,IMIN,(DIST(I,L,M),I=1,NCL)    PRINT INFO
00157   87      CONTINUE                                NEXT COMP.
00158           NFALS = NIM(L)-NCORR                    GET NUMBER OF ERRORS
00159           NIMM = NIM(L)
00160           PER = 100.*FLOAT(NCORR)/FLOAT(NIMM)     SUCCESS PERC.
00161           TYPE 152,NCORR,NFALS,PER                TYPE INFO
00162           PRINT 152,NCORR,NFALS,PER
00163   90      CONTINUE                                NEXT CLASS
00164   C
00165   C       END OF PROGRAM
00166   C
00167   C
00168   C       FORMAT BLOCK
```

```
CLASS2  CLASS2.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:43   PAGE 1-3


00169   C
00170   99      FORMAT (//,T10,'THIS PROGRAM WILL CLASSIFY YOUR TRAINING
00171           1 PATTERNS',/,T10,'USING A MINIMUM DISTANCE CLASSIFIER',
00172           2/,T10,'SOME INFORMATION IS REQUIRED; ANSWER QUESTIONS,
00173           3 FINISHING BY CR',//)
00174   100     FORMAT (T10,'ENTER NUMBER OF CLASSES : ',$)
00175   101     FORMAT (T10,'ENTER TOTAL NUMBER OF COMPONENTS : ',$)
00176   102     FORMAT (T10,'ENTER NUMBER OF PATTERNS IN THIS SET : ',$)
00177   103     FORMAT (T10,'ENTER FILENAME OF TRAINING SET OF CLASS',
00178           1I3,' : ',$)
00179   104     FORMAT (T10,'ENTER FILENAME OF INV. COV. MATRIX : ',$)
00180   105     FORMAT (T10,'DO YOU WANT TO USE ALL COMP. (YES=1,
00181           1NO=0) ? ',$)
00182   106     FORMAT (T10,'TYPE IN THE DESIRED COMPONENTS NUMBERS,
00183           1 ONE AT A TIME.',/,T10,'WHEN FINISHED, TYPE A ZERO.',/)
00184   107     FORMAT (T10,'TYPE COMP. # : ',$)
00185   109     FORMAT (T10,'TYPE IN DIMENSION OF INV. MATRIX :  ',$)
00186   C
00187   150     FORMAT (///,T30,'RESULTS FOR CLASS',I3,//)
00188   153     FORMAT (T2,'CLASSIF.:',20X,'DISTANCES TO CLASSES:',/,
00189           2T10,' CLASS 1     CLASS 2     CLASS 3     CLASS 4     CLASS
00190           3 5     CLASS 6     CLASS 7     CLASS 8     CLASS 9
00191           4     CLASS 10',//)
00192   151     FORMAT (T3,I3,2X,10(E10.3,2X))
00193   152     FORMAT (/,T10,'--NUMBER OF CORRECT CLASSIF.   :',I3,/,
00194           1T10,'  NUMBER OF ERRONEOUS CLASSIF. :',I3,/,T10,
00195           2'  PERCENTAGE OF SUCCESS      :',F10.5,/)
00196   C
00197   200     FORMAT (I3)
00198   201     FORMAT (A10)
00199   202     FORMAT (E14.7)
00200   C
00201   C       END OF FORMAT BLOCK
00202   C       AND END OF PROGRAM
00203   C
00204           END


SUBPROGRAMS CALLED

FLOAT.


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1        MEAN    2       *N      135     *K      136     *NCL    137
*PROD    140      *NFALS  141     *NDM    142     *IMIN   143      D2     144
 NIM     170       V      202     *M      13502   *J      13503    .S0007 13504
 .S0006  13505     D1     13506    .S0005 13532   *NIMM   13533    .S0004 13534
 .S0003  13535     .S0002 13536    DIST   13537    .S0001 30137    .S0000 30140
*NMB     30141     .S0017 30142    IND    30143    .S0016 30276    .S0015 30277
 .S0014  30300     .S0013 30301    .S0012 30302    .S0011 30303    .S0010 30304
*PER     30305    *L      30306   *NYN    30307   *I      30310   *VMIN   30311
 C       30312    *NCORR  50443   *NN     50444
```

```
CLASS2  CLASS2.FOR      FORTRAN V.5(515) /KI    26-JUL-77       17:43   PAGE 1-4


TEMPORARIES

 .00000 50767    .00001 50770

CLASS2    NO ERRORS DETECTED
```

```
THERE ARE 11 COMMANDS AVAILABLE  :

- 1-    PARTIAL INITIALIZATION (REJECT ALL COMPONENTS)
- 2-    SELECT SPECIFIED COMPONENT
- 3-    REJECT SPECIFIED COMPONENT
- 4-    SELECT THE NEXT BEST COMPONENT(S) -FOR A SPECIFIED NUMBER
        OF STEPS- UNLESS THE CONTRIBUTION IS BELOW THE THRESHOLD-
        TO-SELECT
- 5-    REJECT THE NEXT POOREST COMPONENT(S) -FOR A SPECIFIED NUMBER
        OF STEPS- UNLESS THE CONTRIBUTION IS BELOW THE THRESHOLD-
        TO-REJECT
- 6-    SET THE THRESHOLD-TO-SELECT (REAL VALUE)
- 7-    SET THE THRESHOLD-TO-REJECT (REAL VALUE)
- 8-    PRINT A LINE OF SEPARATION
- 9-    TYPE STATUS TO THE USER'S TERMINAL
-10-    PRINT STATUS
-11-    END OF PROGRAM
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C       THIS PROGRAM USES WILKS MATRICES TO REPRESENT A
00003   C       MULTICLASS PATTERN RECOGNITION PROBLEM WITH
00004   C       THE FIRST TWO STATISTICAL MOMENTS
00005   C       THE SELECTION OF COMPONENTS IS DONE ACCORDING
00006   C       TO THE FOLLOWING CRITERIA
00007   C       AT ANY STAGE, IF SC ASKED, REJECT THE COMPONENT
00008   C        WITH THE SMALLEST CONTRIBUTION, UNLESS THIS CON-
00009   C        TRIBUTION IS ABOVE A SPECIFIED THRESHOLD-TO-REMOVE
00010   C       AT ANY STAGE, IF SC ASKED, ADD THE COMPONENT
00011   C        WITH THE LARGEST CONTRIBUTION, UNLESS THIS CON-
00012   C        TRIBUTION IS BELOW A SPECIFIED THRESHOLD-TO-ADD
00013   C
00014   C       THE COMMANDS ALSO ALLOW TO ADD OR A REJECT A SPECIFIED
00015   C       COMPONENT, TO PRINT OR TYPE A BRIEF STATUS.
00016   C       THE SEARCH IS INITIATED FOR A SPECIFIED NUMBER OF STEPS
00017   C
00018           PROGRAM SELEC1
00019           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00020
00021   C
00022   C       THESE PARAMETERS DEFINE UPPER BOUNDS ON THE NUMBER
00023   C       OF COMPONENTS AND THE NUMBER OF CLASSES
00024   C
00025           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00026           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00027           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00028           DOUBLE PRECISION FN(1:NCLASS),W1,W2,M
00029           REAL CONT
00030           INTEGER IND,NMB
00031           DATA IND/NCOMP*0/,CONT/NCOMP*0/
00032   C
00033   C       W1 WILL CONTAIN THE POOLED COVARIANCE MATRIX
00034   C       W2 WILL CONTAIN THE VARIABLE INVERSE OF W
00035   C       M WILL CONTAIN THE MEAN VECTOR OF EACH CLASS
00036   C       IND WILL KEEP TRACE OF THE SELECTED COMPONENTS
00037   C       CONT WILL STORE THE INDIVIDUAL CONTRIBUTION OF
00038   C             EACH COMPONENT
00039   C       NMB WILL CONTAIN THE NUMBER OF PATTERNS IN EACH CLASS
00040   C       FN WILL STORE THE NAMES OF THE COVARIANCE MATR. FILES
00041   C
00042           TYPE 100                                MESSAGE TO USER
00043   1       TYPE 101                                ASK FOR NUMBER OF CLASSES
00044           ACCEPT 200,NCL
00045           IF (NCL .LE. 0 .OR. NCL .GT. NCLASS) GO TO 1
00046   2       TYPE 102                                ASK FOR NUMBER OF COMP.
00047           ACCEPT 200,NCO
00048           IF (NCO .LE. 0 .OR. NCO .GT. NCOMP) GO TO 2
00049   C
00050   C       BOTH ANSWERS CHECKED
00051   C
00052           DO 10 I=1,NCL
00053           TYPE 103,I                              ASK FILENAME OF I-TH MATR.
00054           ACCEPT 201,FN(I)
00055   5       TYPE 104                                ASK FOR NUMBER OF IM. IN CLASS
00056           ACCEPT 200,NMB(I)
```

```
SELEC1  SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1-1


00057           IF (NMB(I) .LE. 0 .OR. NMB(I) .GT. NIMAG) GO TO 5
00058   10      CONTINUE
00059   C
00060   C       PARAMETERS INPUT FINISHED
00061   C
00062   C       THE PROGRAM WILL NOW GET THE W1 AND M MATRICES
00063   C       AND INITIALIZE THE W2 MATRIX
00064   C
00065           DO 15 I=1,NCO
00066           DO 15 J=1,NCO
00067   15      W1(I,J) = 0.
00068   C
00069           DO 30 K=1,NCL
00070           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=FN(K))
00071           READ (20,202) (M(I,K),I=1,NCO)    READ IN K-TH MEAN VECTOR
00072           READ (20,202) ((W2(I,J),I=1,NCO),J=1,NCO)
00073   C
00074   C       THIS TEMPORARY STORAGE IS USED TO PROGRESSIVELY FORM W1
00075   C
00076           DO 20 I=1,NCO
00077           DO 20 J=1,NCO
00078   20      W1(I,J) = W1(I,J)+W2(I,J)*NMB(K)
00079   C
00080   C       THE PROCESS IS REPEATED FOR ALL CLASSES
00081   C
00082           CLOSE (UNIT=20,DEVICE='DSK',FILE=FN(K))
00083   30      CONTINUE
00084   C
00085           DO 35 I=1,NCO
00086           DO 35 J=1,NCO
00087   35      W2(I,J) = 0.
00088   C
00089           TA = 0.
00090           TB = 0.                         INITIALIZE THRESHOLDS
00091           WAL = 0.                        INITIALIZE WILKS' MEASURE
00092   C
00093   C       THE INTIALIZATION IS NOW COMPLETE
00094   C       THE PROGRAM ENTERS THE COMMAND LOOP
00095   C
00096           TYPE 105                        MESSAGE TO USER
00097   50      TYPE 106                        ASK FOR COMMAND
00098           ACCEPT 200,NCM                  GET IT
00099           IF (NCM .LE. 0 .OR. NCM .GT. 11) GO TO 50
00100   C
00101   C       THE COMMAND IS VALID, SO EXECUTE
00102   C
00103           GO TO (1000,1100,1200,1300,1400,1500,1600,1700,
00104          1,1800,1900,2000)NCM             NUMERICAL SWITCH AS COMMAND INTE
00105   C
00106   C
00107   C       FIRST COMMAND : REJECT ALL COMPONENTS
00108   C
00109   1000    DO 1005 I=1,NCO
00110           IND(I) = 0
00111           DO 1005 J=1,NCO
00112   1005    W2(I,J) = 0.
```

```
00113            WAL = 0.
00114            GO TO 50                    NEXT COMMAND
00115    C
00116    C
00117    C       SECOND COMMAND : ADD A SPECIFIED COMPONENT
00118    C
00119    1100    TYPE 107                    ASK FOR COMPONENT #
00120            ACCEPT 200,NCM
00121            IF (NCM .LE. 0 .OR. NCM .GT. NCO) GO TO 1100
00122            IF (IND(NCM) .NE. 0) GO TO 1150  ALREADY EXISTING
00123    C
00124    C       COMPONENT NUMBER VALID
00125    C
00126            IND(NCM) = NCM
00127            CALL DISTUP(NCM,NCO,NCL,WAL)     UPDATE WILKS' MEASURE
00128            CALL INVUP(NCM,NCO,NCL)          UPDATE INVERSE MATRIX
00129            GO TO 50                         NEXT COMMAND
00130    C
00131    1150    TYPE 150                         COMPONENT ALREADY SELECTED
00132            GO TO 50                         NEXT COMMAND
00133    C
00134    C
00135    C       THIRD COMMAND
00136    C
00137    1200    TYPE 108                         ASK FOR COMPONENT #
00138            ACCEPT 200,NCM
00139            IF (NCM .LE. 0 .OR. NCM .GT. NCO) GO TO 1200
00140            IF (IND(NCM) .EQ. 0) GO TO 1250  ALREADY REJECTED
00141    C
00142    C       COMPONENT NUMBER VALID
00143    C
00144            IND(NCM) = 0
00145            CALL DISTDN(NCM,NCO,NCL,WAL)     UPDATE WILKS' MEASURE
00146            CALL INVDN(NCM,NCO,NCL)          UPDATE INVERSE MATRIX
00147            GO TO 50                         NEXT COMMAND
00148    C
00149    1250    TYPE 151                         ALREADY REJECTED
00150            GO TO 50
00151    C
00152    C
00153    C       FOURTH COMMAND : SELECT THE NEXT BEST COMPONENT
00154    C
00155    1300    TYPE 109                         ASK FOR NUMBER OF STEPS
00156            ACCEPT 200,NSTEP
00157            IF (NSTEP .LE. 0 .OR. NSTEP .GT. NCO) GO TO 1300
00158            DO 1350 L=1,NSTEP
00159            CALL UPCON(NCO,NCL)              GET CONTRIBUTIONS
00160            CALL MAX(NCO,K)                  RETURNS INDEX IN K
00161            IF (K .EQ. 0) GO TO 1390         ALL COMP. ALREADY SEL.
00162            RATIO = 0.
00163            IF (WAL .EQ. 0.) GO TO 1320
00164            RATIO = 100.*CONT(K)/WAL
00165            IF (RATIO .LT. TA) GO TO 1370    THRESHOLD REACHED
00166    1320    IND(K) = K                       ADD COMPONENT
00167            TYPE 160,K,RATIO
00168            PRINT 160,K,RATIO                INFORMATION
```

```
SELEC1  SELEC1.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1-3


00169           CALL DISTUP(K,NCO,NCL,#AL)          UPDATE WILKS' MEASURE
00170           CALL INVUP(K,NCO,NCL)               UPDATE INVERSE MATRIX
00171   1350    CONTINUE                            NEXT STEP
00172           GO TO 50                            NEXT COMMAND
00173   C
00174   1370    TYPE 152                            THRESHOLD-TO-ADD REACHED
00175           PRINT 152
00176           GO TO 50                            NEXT COMMAND
00177   C
00178   1390    TYPE 154
00179           PRINT 154                           ERROR, ALL COMP. SEL.
00180           GO TO 50                            NEXT COMMAND
00181   C
00182   C
00183   C       FIFTH COMMAND : REJECT NEXT #ORST COMPONENT
00184   C
00185   1400    TYPE 109                            ASK FOR NUMBER OF STEPS
00186           ACCEPT 200,NSTEP
00187           IF (NSTEP .LE. 0 .OR. NSTEP .GT. NCO) GO TO 1400
00188           DO 1450 L=1,NSTEP
00189           CALL DNCON(NCO,NCL)                 GET CONTRIBUTIONS
00190           CALL MIN(NCO,K)                     RETURNS INDEX IN K
00191           IF (K .EQ. 0) GO TO 1490            ALL COMP. ALREADY REJECTED
00192           RATIO = 100.*CONT(K)/#AL
00193           IF (RATIO .GT. TR) GO TO 1470       THRESHOLD-TO-REJECT REACHED
00194           IND(K) = 0                          REJECT COMPONENT
00195           TYPE  161,K,RATIO
00196           PRINT 161,K,RATIO                   INFORMATION
00197           CALL DISTDN(K,NCO,NCL,#AL)          UPDATE WILKS' MEASURE
00198           CALL INVDN(K,NCO,NCL)               UPDATE INVERSE MATRIX
00199   1450    CONTINUE                            NEXT STEP
00200           GO TO 50                            NEXT COMMAND
00201   C
00202   1470    TYPE 163                            THRESHOLD-TO-REJECT REACHED
00203           PRINT 163
00204           GO TO 50                            NEXT COMMAND
00205   C
00206   1490    TYPE 165
00207           PRINT 165                           ERROR, ALL COMP. REJECTED
00208           GO TO 50                            NEXT COMMAND
00209   C
00210   C
00211   C       SIXTH COMMAND : SET THRESHOLD-TO-ADD
00212   C
00213   1500    TYPE 110                            ASK FOR THRESHOLD-TO-ADD
00214           ACCEPT 203,TA
00215           IF (TA .LT. 0.) GO TO 1500
00216           GO TO 50                            NEXT COMMAND
00217   C
00218   C
00219   C       SEVENTH COMMAND : SET THRESHOLD-TO-REJECT
00220   C
00221   1600    TYPE 111                            ASK FOR THRESHOLD-TO-REJECT
00222           ACCEPT 203,TR
00223           IF (TR .LT. 0.) GO TO 1600
00224           GO TO 50                            NEXT COMMAND
```

```
SELEC1  SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1-4


00225   C
00226   C
00227   C       EIGHTH  COMMAND : PRINT A LINE OF SEPARATION
00228   C
00229   1700    PRINT 170                               PRINT A SEPARATION LINE
00230           GO TO 50                                NEXT COMMAND
00231   C
00232   C
00233   C       NINTH COMMAND : TYPE STATUS
00234   C
00235   1800    NB = 0
00236           DO 1805 I=1,NCU
00237           IF (IND(I) .NE. 0) NB=NB+1
00238   1805    CONTINUE
00239           TYPE 162,NB,WAL,TA,TR
00240           CALL SNB(NCO)                           JUSTIFY NUMBERS OF SEL.
00241                                                   COMP. TO THE LEFT OF CONT
00242           TYPE 163,(CONT(I),I=1,NB)
00243           TYPE 164
00244           GO TO 50                                NEXT COMMAND
00245   C
00246   C
00247   C       TENTH COMMAND : PRINT STATUS
00248   C
00249   1900    NB = 0
00250           DO 1905 I=1,NCU
00251           IF (IND(I) .NE. 0) NB=NB+1
00252   1905    CONTINUE
00253           PRINT 162,NB,WAL,TA,TR
00254           CALL SNB(NCO)
00255           PRINT 163,(CONT(I),I=1,NB)
00256           PRINT 171
00257           GO TO 50                                NEXT COMMAND
00258   C
00259   C
00260   C       ELEVENTH COMMAND : END OF PROGRAM
00261   C
00262   2000    STOP 'NORMAL ENDING'
00263   C
00264   C
00265   C       FORMAT BLOCK
00266   C
00267   100     FORMAT (///,T10,'THIS PROGRAM WILL HELP YOU SELECT MEA
00268           1SUREMENTS.',/,T10,'PLEASE ANSWER THE FOLLOWING QUESTI
00269           2ONS, FINISHING BY CR.',///)
00270   101     FORMAT (T10,'ENTER NUMBER OF CLASSES : ',$)
00271   102     FORMAT (T10,'ENTER NUMBER OF COMPON. : ',$)
00272   103     FORMAT (T10,'ENTER FILENAME FOR CLASS',I3,' : ',$)
00273   104     FORMAT (T10,'ENTER # OF IMAGES IN THIS CLASS : ',$)
00274   105     FORMAT (/,T10,'THE INITIALIZATION IS NOW FINISHED. YOU
00275           1 ENTER THE COMMAND LOOP.',//)
00276   106     FORMAT (T10,'ENTER YOUR COMMAND (1-11) : ',$)
00277   107     FORMAT (T10,'ENTER # OF COMP. TO BE ADDED : ',$)
00278   108     FORMAT (T10,'ENTER # OF COMP. TO BE REJECTED : ',$)
00279   109     FORMAT (T10,'ENTER # OF STEPS TO BE PERFORMED : ',$)
00280   110     FORMAT (T10,'ENTER THRESHOLD-TO-ADD (IN %) : ',$)
```

```
SELEC1  SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1-5


00281   111     FORMAT (T10,'ENTER THRESHOLD-TO-REJECT (IN %) : ',$)
00282   C
00283   150     FORMAT (T10,'COMPONENT ALREADY SELECTED  ',/)
00284   151     FORMAT (T10,'COMPONENT ALREADY REJECTED  ',/)
00285   152     FORMAT (T10,'THRESHOLD-TO-ADD REACHED  ',/)
00286   153     FORMAT (T10,'THRESHOLD-TO-REJECT REACHED  ',/)
00287   154     FORMAT (T10,'ALL COMPONENTS ALREADY SELECTED',/)
00288   155     FORMAT (T10,'ALL COMPONENTS ALREADY REJECTED',/)
00289   C
00290   160     FORMAT (T10,'THE SELECTED COMPONENT IS NUMBER',I3,
00291           1/,T15,'WITH A CONTRIBUTION OF ',E9.2,' %',/)
00292   161     FORMAT (T10,'THE REJECTED COMPONENT IS NUMBER',I3,
00293           1/,T15,'WITH A DIMINUTION OF ',E9.2,' %',/)
00294   162     FORMAT (T10,'THE PRESENT NUMBER OF COMPONENTS IS',I3,
00295           1/,T10,'THE INTERCLASS DISTANCE MEASURE IS ',E11.4,/,
00296           2T10,'THE THRESHOLD-TO-ADD IS    ',E11.4,/,
00297           3T10,'THE THRESHOLD-TO-REJECT IS ',E11.4,/,
00298           4T10,'THE SELECTED COMPONENTS ARE :',/)
00299   163     FORMAT (10(4X,F3.0))
00300   164     FORMAT (/)
00301   C
00302   170     FORMAT (/,80('*'),/)
00303   171     FORMAT (//)
00304   C
00305   200     FORMAT (I3)'
00306   201     FORMAT (A10)
00307   202     FORMAT (D22.15)
00308   203     FORMAT (F8.3)
00309   C
00310   C       END OF FORMAT BLOCK
00311   C
00312           END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED

MIN     UPCON   DISTUP  MAX     DNCON   DISTDN
INVDN   INVUP   SNB

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "$" NOT REFERENCED

 .S0020 1       *NCU    2       *TB     3       *NB     4       *K      5
*NCL    6       *TR     7       *TA     10      *RATIO  11      *NCM    12
*WAL    13      *J      14       .S0007 15       .S0006 16       .S0005 17
 .S0004 20       .S0003 21       .S0002 22       .S0001 23       .S0000 24
 .S0017 25       FN     26       .S0016 76       .S0015 77       .S0014 100
 .S0013 101      .S0012 102      .S0011 103      .S0010 104      *L     105
*NSTEP  106     *I      107      .S0022 110      .S0021 111
```

```
SELEC1  SELEC1.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1-6

TEMPORARIES

 .00000 532       .00001 533

SELEC1    NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MAX(NCO,K)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013   C
00014           K = 0
00015           VMAX = -1.E+26
00016           DO 10 I=1,NCO
00017           IF (IND(I) .NE. 0) GO TO 10       ONLY UNSEL. COMP.
00018           IF (VMAX .GE. CONT(I)) GO TO 10   SEARCH A LARGER
00019           K = I
00020           VMAX = CONT(I)                    FOUND A LARGER
00021   10      CONTINUE
00022           RETURN
00023           END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *VMAX   2       *K      3       .S0000.4        *I      5

TEMPORARIES


MAX       NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       14:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MIN(NCO,K)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013   C
00014           K = 0
00015           VMIN = 1.E+26
00016           DO 10 I=1,NCO
00017           IF (IND(I) .EQ. 0) GO TO 10       ONLY UNSEL. COMP.
00018           IF (VMIN .LE. CONT(I)) GO TO 10   SEARCH A SMALLER
00019           K = I
00020           VMIN = CONT(I)                    FOUND A SMALLER
00021   10      CONTINUE
00022           RETURN
00023           END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *K      2       .50000 3        *I      4       *VMIN   5

TEMPORARIES


MIN       NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE SNB(NCO)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013   C
00014           DO 5 I=1,NCO
00015   5       CONT(I) = 0.
00016           J = 0
00017           DO 10 I=1,NCO
00018           IF (IND(I) .EQ. 0) GO TO 10
00019           J = J+1
00020           CONT(J) = I
00021   10      CONTINUE
00022           RETURN
00023           END                             NUMBERS PACKED TO THE LEFT

COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "+" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

+NCO    1       +J      2       .50001 3        .50000 4        +I      5

TEMPORARIES


SNB       NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE UPCON(NCO,NCL)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013           DOUBLE PRECISION TEMP,BETA,Z(1:NCOMP)
00014   C
00015           DO 5 I=1,NCO
00016   5       CONT(I) = 0.
00017   C
00018           DO 100 K=1,NCO                          GENERAL LOOP FOR COMP.
00019   C
00020   C       THSI LOOP WILL COMPUTE THE INDIVIDUAL CONTRIBUTION OF
00021   C       EACH UNSELECTED COMPONENT
00022   C
00023           IF (IND(K) .NE. 0) GO TO 100            ONLY UNSEL. COMP.
00024   C
00025   C       COMPUTE THE VECTOR PRODUCT W2*W1(K)
00026   C
00027           DO 20 I=1,NCO
00028           Z(I) = 0.
00029           IF (IND(I) .EQ. 0) GO TO 20             ONLY SEL. COMP.
00030           DO 20 J=1,NCO
00031           IF (IND(J) .EQ. 0) GO TO 20
00032           Z(I) = Z(I)+W2(I,J)*W1(J,K)
00033   20      CONTINUE
00034   C
00035   C       FORM THE SCALAR PRODUCT Z*W(K)
00036   C
00037           TEMP = 0.
00038           DO 30 I=1,NCO
00039           IF (IND(I) .EQ. 0) GO TO 30             ONLY SEL. COMP.
00040           TEMP = TEMP+Z(I)*W1(K,I)
00041   30      CONTINUE
00042           BETA = W1(K,K)-TEMP
00043   C
00044   C       TEST FOR POSITIVE DEFINITENESS
00045   C
00046           IF (BETA .GT. 0.) GO TO 150             ERROR, NOT POS.DEF.
00047           TYPE 200                                SO TYPE A WARNING
00048   200     FORMAT (T10,'WARNING : NO MORE POS.DEF.',/)
00049   C
00050   C       COMPUTE THE CONTRIBUTION ; ALL MEANS MUST BE TAKEN INTO
00051   C       ACCOUNT
00052   C
00053   150     DO 50 L=1,NCL
00054   C
00055   C       FORM THE SCALAR PRODUCT M(L)*Z
00056   C
```

```
UPCON   SELEC1.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1-1


00057           TEMP = 0.
00058           DO 40 I=1,NCO
00059           IF (IND(I) .EQ. 0) GO TO 40       ONLY SEL. COMP.
00060           TEMP = TEMP+Z(I)*M(I,L)
00061   40      CONTINUE
00062           TEMP = (TEMP-M(K,L))**2
00063   C
00064   C       NOW ADD THIS MEAN'S CONTR. TO THE RUNNING VALUE
00065   C
00066           CONT(K) = CONT(K)+NMB(L)*TEMP/BETA
00067   50      CONTINUE                          NEXT MEAN
00068   C
00069   100     CONTINUE                          NEXT COMPONENT
00070           RETURN
00071   C
00072           END


COMMON BLOCKS

/.COMM./(+44156)
#1      +0      +2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       Z       2       BETA    200     *K      202     *NCL    203
*J      204     .S0006  205     .S0005  206      .S0004 207      .S0003 210
 .S0002 211     .S0001  212     .S0000  213     TEMP    214     *L      216
*I      217

TEMPORARIES

 .A0016 230

UPCON     NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE ONCON(NCO,NCL)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013           DOUBLE PRECISION TEMP
00014   C
00015           DO 5 I=1,NCO
00016   5       CONT(I) = 0.
00017   C
00018           DO 100 K=1,NCO              GENERAL LOOP FOR COMP.
00019   C
00020   C       THIS LOOP WILL COMPUTE THE INDIVIDUAL CONTRIBUTION OF
00021   C       EACH SEL. COMP. TO THE WILKS' MEASURE
00022   C
00023           IF (IND(K) .EQ. 0) GO TO 100      ONLY SELECTED COMP.
00024   C
00025   C       COMPUTE THE CONTRIBUTION; ALL MEANS MUST BE TAKEN INTO
00026   C       ACCOUNT
00027   C
00028           DO 50 L=1,NCL
00029   C
00030   C       FORM THE SCALAR PRODUCT M(L)*W2(K)
00031   C
00032           TEMP = 0.
00033           DO 40 I=1,NCO
00034           IF (IND(I) .EQ. 0) GO TO 40       ONLY SEL. COMP.
00035           IF (I .EQ. K) GO TO 40            SKIP THIS INDEX
00036           TEMP = TEMP+M(I,L)*W2(I,K)
00037   40      CONTINUE
00038           TEMP = (TEMP+W2(K,K)*M(K,L))**2
00039           CONT(K) = CONT(K)+TEMP*NMB(L)/W2(K,K)
00040   50      CONTINUE
00041   C
00042   C       THE CONTRIBUTION OF THE K=TH COMP. IS NOW COMPUTED
00043   C
00044   100     CONTINUE                          NEXT COMPONENT
00045   C
00046           RETURN
00047           END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132
```

```
DNCON   SELEC1.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1-1


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1       *K       2       *NCL     3       .S0003 4       .S0002 5
 .S0001 6        .S0000 7        TEMP    10      *L       12      *1       13

TEMPORARIES


DNCON      NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE DISTUP(K,NCO,NCL,AL)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013           DOUBLE PRECISION TEMP,BETA,Z(1:NCOMP)
00014   C
00015   C       COMPUTE THE VECTOR PRODUCT W2*W1(K)
00016   C
00017           DO 20 I=1,NCO
00018           Z(I) = 0.
00019           IF (IND(I) .EQ. 0) GO TO 20        ONLY SEL. COMP.
00020           IF (I .EQ.  K) GO TO 20
00021           DO 20 J=1,NCO
00022           IF (IND(J) .EQ. 0) GO TO 20
00023           IF (J .EQ. K) GO TO 20
00024           Z(I) = Z(I)+W2(I,J)*W1(J,K)
00025   20      CONTINUE
00026   C
00027   C       FORM THE SCALAR PRODUCT Z*W(K)
00028   C
00029           TEMP = 0.
00030           DO 30 I=1,NCO
00031           IF (IND(I) .EQ. 0) GO TO 30        ONLY SEL. COMP.
00032           IF (I .EQ. K) GO TO 30
00033           TEMP = TEMP+Z(I)*W1(I,K)
00034   30      CONTINUE
00035           BETA = W1(K,K)-TEMP
00036   C
00037   C       TEST FOR POSITIVE DEFINITENESS
00038   C
00039           IF (BETA .GT. 0.) GO TO 150        ERROR, NOT POS.DEF.
00040           TYPE 200                           SO TYPE A WARNING
00041   200     FORMAT (T10,'WARNING : NO MORE POS.DEF.',/)
00042   C
00043   C       COMPUTE THE CONTRIBUTION ; ALL MEANS MUST BE TAKEN INTO
00044   C       ACCOUNT
00045   C
00046   150     DO 50 L=1,NCL
00047   C
00048   C       FORM THE SCALAR PRODUCT M(L)*Z
00049   C
00050           TEMP = 0.
00051           DO 40 I=1,NCO
00052           IF (IND(I) .EQ. 0) GO TO 40        ONLY SEL. COMP.
00053           IF (I .EQ. K) GO TO 40
00054           TEMP = TEMP+Z(I)*M(I,L)
00055   40      CONTINUE
00056           TEMP = (TEMP-M(K,L))**2
```

```
DISTUP  SELEC1.FOR      FORTRAN V.S(515) /KI    27-JUL-77       19:50   PAGE 1-1


00057   C
00058   C       NOW ADD THIS MEAN'S CONTR. TO THE RUNNING VALUE
00059   C
00060           #AL = #AL+NM#(L)*TEMP/BETA
00061   50      CONTINUE                                NEXT MEAN
00062   C
00063           RETURN
00064   C
00065           END


COMMON BLOCKS

/.COMM./(+44156)
#1      +0      #2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS    "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       Z       2       BETA    200     *K      202     *NCL    203
*#AL    204     *J      205     .S0004  206      .S0003 207      .S0002 210
 .S0001 211      .S0000 212      TEMP    213     *L      215     *I      216


TEMPORARIES

 .A0016 227

DISTUP    NO ERRORS DETECTED
```

```
MAIN.   SELECT.FOR      FORTRAN V.3(515) /KI    27-JUL-77       19:50   PAGE 1


00001    C
00002    C
00003    C
00004            SUBROUTINE DISTDN(K,NCO,NCL,WAL)
00005            PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006    C
00007            COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008            COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009            COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010            DOUBLE PRECISION W1,W2,M
00011            REAL CONT
00012            INTEGER IND,NMB
00013            DOUBLE PRECISION TEMP
00014    C
00015    C
00016    C       COMPUTE THE CONTRIBUTION: ALL MEANS MUST BE TAKEN INTO
00017    C       ACCOUNT
00018    C
00019            DO 50 L=1,NCL
00020    C
00021    C       FORM THE SCALAR PRODUCT M(L)*W2(K)
00022    C
00023            TEMP = 0.
00024            DO 40 I=1,NCO
00025            IF (IND(I) .EQ. 0) GO TO 40        ONLY SEL. COMP.
00026            IF (I .EQ. K) GO TO 40             SKIP THIS INDEX
00027            TEMP = TEMP+M(I,L)*W2(I,K)
00028    40      CONTINUE
00029            TEMP = (TEMP+W2(K,K)*M(K,L))**2
00030            WAL = WAL-TEMP*NMB(L)/W2(K,K)
00031    50      CONTINUE
00032    C
00033    C       THE MEASURE IS NOW UPDATED
00034    C
00035            RETURN
00036            END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      +2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *K      2       *NCL    3       *WAL    4       .S0001 5
 .S0000 6       TEMP    7       *L      11      *I      12
```

```
DISTDN  SELEC1.FOR      FORTRAN V.5(515) /K1    27-JUL-77       19:50   PAGE 1-1

TEMPORARIES

DISTDN    NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE INVUP(K,NCO,NCL)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013           DOUBLE PRECISION Z(1:NCOMP),BETA,TEMP
00014   C
00015   C
00016   C       COMPUTE THE VECTOR PRODUCT W2*W1(K)
00017   C
00018           DO 10 I=1,NCO
00019           Z(I) = 0.
00020           IF (IND(I) .EQ. 0) GO TO 10
00021           IF (I .EQ. K) GO TO 10
00022           DO 10 J=1,NCO
00023           IF (IND(J) .EQ. 0) GO TO 10
00024           IF (J .EQ. K) GO TO 10
00025           Z(I) = Z(I)+W2(I,J)*W1(J,K)
00026   10      CONTINUE
00027   C
00028   C       FORM THE SCALAR PRODUCT Z*W(K)
00029   C
00030           TEMP = 0.
00031           DO 30 I=1,NCO
00032           IF (IND(I) .EQ. 0) GO TO 30
00033           IF (I .EQ. K) GO TO 30
00034           TEMP = TEMP+Z(I)*W1(I,K)
00035   30      CONTINUE
00036           BETA = W1(K,K)-TEMP
00037   C
00038   C       TEST FOR POSITIVE-DEFINITENESS
00039   C
00040           IF (BETA .GT. 0.) GO TO 150       ERROR, NOT POSITIVE-DEFINITE
00041           TYPE 200                          SO TYPE A WARNING
00042   200     FORMAT (T10,'WARNING : NO MORE POS.DEF.',/)
00043   C
00044   150     W2(K,K) = 1./BETA
00045           DO 40 I=1,NCO
00046           IF (IND(I) .EQ. 0) GO TO 40
00047           IF (I .EQ. K) GO TO 40
00048           W2(I,K) = -Z(I)/BETA
00049           W2(K,I) = W2(I,K)
00050   40      CONTINUE
00051   C
00052   C       THE NEW ROW AND THE NEW COLUMN  OF THE INVERSE HAVE
00053   C       BEEN UPDATED. THE OLD BODY WILL NOW BE UPDATED.
00054   C
00055           DO 50 I=1,NCO
00056           IF (IND(I) .EQ. 0) GO TO 50
```

```
INVUP  SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1-1


00057            IF (I .EQ. K) GO TO 50
00058            DO 50 J=I+NCO
00059            IF (IND(J) .EQ. 0) GO TO 50
00060            IF (J .EQ. K) GO TO 50
00061            *2(I,J) = W2(I,J)+Z(I)*Z(J)/BETA
00062            W2(J,I) = W2(I,J)
00063   50       CONTINUE
00064            RETURN
00065   C
00066            END


COMMON BLOCKS

/.COMM./(+44156)
W1       +0       *2       +17402   *1       +37004   CONT     +43734   IND      +44033
NMB      +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1        Z        2        BETA     200      *K       202      %NCL
*J       203      .S0005 204        .S0004 205        .S0003 206        .S0002 207
 .S0001 210       .S0000 211       TEMP     212      *I       214

TEMPORARIES

 .A0016 225

INVUP      NO ERRORS DETECTED
```

```
MAIN.   SELEC1.FOR      FORTRAN V.5(515) /KI    27-JUL-77       19:50   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE INVON(K,NCO,NCL)
00005           PARAMETER NCOMP=63,NCLASS=20,NIMAG=100
00006   C
00007           COMMON W1(1:NCOMP,1:NCOMP),W2(1:NCOMP,1:NCOMP)
00008           COMMON M(1:NCOMP,1:NCLASS),CONT(1:NCOMP)
00009           COMMON IND(1:NCOMP),NMB(1:NCLASS)
00010           DOUBLE PRECISION W1,W2,M
00011           REAL CONT
00012           INTEGER IND,NMB
00013   C
00014           DO 20 I=1,NCO
00015           IF (IND(I) .EQ. 0) GO TO 20
00016           IF (I .EQ. K) GO TO 20
00017           DO 20 J=I,NCO
00018           IF (IND(J) .EQ. 0) GO TO 20
00019           IF (J .EQ. K) GO TO 20
00020           W2(I,J) = W2(I,J)-W2(I,K)*W2(J,K)/W2(K,K)
00021           W2(J,I) = W2(I,J)
00022   20      CONTINUE
00023   C
00024   C       NOW THE MAIN BODY OF THE INVERSE HAS BEEN UPDATED
00025   C       THE EXCESS ROW AND COLUMN WILL BE ZEROED
00026   C
00027           DO 30 I=1,NCO
00028           W2(I,K) = 0.
00029   30      W2(K,I)  = 0.
00030           RETURN
00031           END


COMMON BLOCKS

/.COMM./(+44156)
W1      +0      W2      +17402  M       +37004  CONT    +43734  IND     +44033
NMB     +44132

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *K      2       %NCL            *J      3       .S0002 4
 .S0001 5        .S0000 6       *I      7

TEMPORARIES


INVON     NO ERRORS DETECTED
```

```
THERE ARE 21 COMMANDS AVAILABLE :

- 1-    PRINT A LINE OF SEPARATION
- 2-    PARTIAL INITIALIZATION (REJECT ALL COMPONENTS, RESET ALL WEIGHTS)
- 3-    SELECT SPECIFIED COMPONENT
- 4-    REJECT SPECIFIED COMPONENT
- 5-    SELECT NEXT BEST COMPONENT(S) USING THE WEIGHTED AVERAGE
       OF ALL INTERCLASS DISTANCES AS CRITERION; THIS IS DONE
       FOR THE SPECIFIED NUMBER OF STEPS, UNLESS THE CONTRIBUTION
       IS BELOW THE THRESHOLD-TO-SELECT
- 6-    REJECT THE NEXT POOREST COMPONENT(S) USING THE SAME CRITERION
       AS ABOVE. THIS IS DONE FOR THE SPECIFIED NUMBER OF STEPS,
       UNLESS THE CONTRIBUTION IS ABOVE THE THRESHOLD-TO-REJECT
- 7-    SELECT THE NEXT BEST COMPONENT(S) USING THE WEIGHTED AVERAGE
       OF ALL INTERCLASS DISTANCES RELATIVE TO A SPECIFIED CLASS
       AS CRITERION; THIS IS DONE FOR THE SPECIFIED NUMBER OF STEPS
       UNLESS THE CONTRIBUTION IS BELOW THE THRESHOLD-TO-SELECT
- 8-    REJECT THE NEXT POOREST COMPONENT(S) USING THE SAME CRITERION
       AS ABOVE. THIS IS DONE FOR THE SPECIFIED NUMBER OF STEPS,
       UNLESS THE CONTRIBUTION IS BELOW THE THRESHOLD-TO-REJECT
- 9-    SELECT THE NEXT BEST COMPONENT(S) USING THE DISTANCE BETWEEN
       TWO SPECIFIED CLASSES AS CRITERION. THIS IS DONE FOR THE
       SPECIFIED NUMBER OF STEPS,  UNLESS THE CONTRIBUTION IS BELOW
       THE THRESHOLD-TO-SELECT
-10-   REJECT THE NEXT POOREST COMPONENT(S) USING THE SAME CRITERION
       AS ABOVE. THIS IS DONE FOR THE SPECIFIED NUMBER OF STEPS,
       UNLESS THE CONTRIBUTION IS ABOVE THE THRESHOLD-TO-REJECT
-11-   TYPE TO THE USER'S TERMINAL THE DISTANCES BETWEEN TWO SPECIFIED
       CLASSES
-12-   TYPE TO THE USER'S TERMINAL THE LIST OF THE SELECTED
       COMPONENTS
-13-   TYPE TO THE USER'S TERMINAL THE SHORT STATUS
       (I.E. NUMBER OF SELECTED COMPONENTS, AVERAGE UNWEIGHTED
       INTERCLASS DISTANCE AND ITS STANDARD DEVIATION, MINIMUM
       DISTANCE AND ITS PAIR OF CLASSES, AND BOTH THRESHOLDS)
-14-   PRINT THE SHORT STATUS, PLUS THE LIST OF THE SELECTED COMPONENTS
-15-   PRINT THE SHORT STATUS, THE LIST OF THE SELECTED COMPONENTS,
       AND THE LIST OF ALL INTERCLASS DISTANCES WITH THEIR WEIGHTS
-16-   SET THRESHOLD-TO-SELECT (A REAL VALUE)
-17-   SET THRESHOLD-TO-REJECT (A REAL VALUE)
-18-   ASSIGN WEIGHT (A REAL VALUE) TO THE DISTANCE BETWEEN A
       SPECIFIED PAIR OF CLASSES
-19-   TYPE TO THE USER'S TERMINAL THE WEIGHT ASSIGNED TO THE DISTANCE
       BETWEEN A SPECIFIED PAIR OF CLASSES
-20-   TYPE TO THE USER'S TERMINAL THE MINIMUM INTERCLASS DISTANCE
       AND ITS PAIR OF CLASSES
-21-   END OF PROGRAM (SAVE CURRENT INVERSE COVARIANCE MATRICES
       ON DISK)
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001    C
00002    C       THIS PROGRAM IS DESIGNED TO PROVIDE INTERACTIVE
00003    C       SELECTION OF FEATURES (= COMPONENTS) FOR A PATTERN
00004    C       RECOGNITION SYSTEM
00005    C       THE INPUT TO THE PROGRAM CONSISTS OF A SET OF FILES,
00006    C       EACH OF WHICH CONTAINS A MEAN VECTOR AND A
00007    C       COVARIANCE MATRIX (AS PRODUCED, TYPICALLY, BY THE
00008    C       ROUTINE STAT.FOR). THE USER MUST SUPPLY THE NUMBER
00009    C       OF THESE FILES, THE DIMENSION OF THE MATRICES, AS WELL
00010    C       AS THE NAMES OF THE FILES.
00011    C       THEN ONE HAS A SET OF 21 COMMANDS AT HIS DISPOSITION,
00012    C       NUMBERED FROM 1 TO 21. A COMMAND IS SELECTED BY TYPING
00013    C       IN ITS CODE NUMBER. SOME COMMANDS REQUIRE ADDITIONAL
00014    C       INFORMATION, THAT WILL BE REQUESTED FROM THE USER AS
00015    C       NECESSARY.
00016    C
00017            PROGRAM SELEC2
00018            PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00019    C
00020    C
00021            COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00022            COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00023            COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00024            REAL CONT,DIST,WEIGHT
00025            INTEGER IND
00026            DOUBLE PRECISION C1,C2,M,Z
00027            DOUBLE PRECISION D(1:NCLASS),D2(1:NCLASS)
00028    C
00029    C       THE ARRAY D CONTAINS THE NAME OF THE FILES
00030    C       C1 CONTAINS THE COVARIANCE MATRICES OF ALL CLASSES
00031    C       C2 CONTAINS THE CURRENT INVERSE COVARIANCE MATRICES
00032    C       M CONTAINS THE MEAN VECTORS OF ALL CLASSES
00033    C       CONT IS USED TO STORE THE INDIVIDUAL CONTRIBUTION OF
00034    C               EACH COMPONENT TO THE TOTAL DISTANCE
00035    C       WEIGHT CONTAINS THE WEIGHT ASSIGNED TO EACH INTERCLASS DISTANCE
00036    C       DIST CONTAINS ALL INTERCLASS DISTANCES
00037    C       IND KEEPS TRACE OF THE SELECTED COMPONENTS.
00038    C       (A ZERO MEANS THAT THE COMPONENT IS NOT SELECTED,
00039    C       ANY OTHER VALUE MEANS THAT THE COMPONENT IS SELECTED)
00040    C
00041    1       TYPE 100
00042            ACCEPT 200,NCL                  GET NUMBER OF CLASSES
00043            IF (NCL .GT. NCLASS .OR. NCL .LE. 0) GO TO 1
00044    2       TYPE 101
00045            ACCEPT 200,NCO                  GET NUMBER OF COMPON.
00046            IF (NCO .GT. NCOMP .OR. NCO .LE. 0) GO TO 2
00047            DO 10 I=1,NCL
00048            TYPE 102,I
00049            ACCEPT 201,D(I)                 GET FILENAME OF CLASS I .
00050            TYPE 118
00051            ACCEPT 201,D2(I)                GET OUTPUT FILENAME
00052    C
00053    C       D2 WILL BE USED AT THE END OF THE PROGRAM, TO WRITE
00054    C       THE COMPUTED COVARIANCE INVERSES TO  DISK
00055    C
00056    10      CONTINUE
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1

00057           DO 15 I=1,NCL
00058           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=O(I))
00059           READ (20,202) (M(J,I),J=1,NCO)
00060   C       READ IN MEAN VECTOR
00061           READ (20,202) ((C1(J,K,I),J=1,NCO),K=1,NCO)
00062   C       READ IN  COV. MATRIX
00063   15      CONTINUE
00064   C
00065   C       THIS LOOP HAS READ THE MEAN VECTORS AND INVERSE
00066   C       COVARIANCE MATRICES IN THE ARRAY C
00067   C
00068           TR = 0.
00069           TA = 0.                                 INITIALIZE THRESH. TO 0
00070           NOIS = NCL*(NCL-1)
00071           TYPE 103                                END OF INITIALIZATION
00072           GO TO 1100                              RESET ALL
00073   C
00074   C       THE INITIALIZATION IS NOW FINISHED, AND THE PROGRAM
00075   C       ENTERS THE COMMAND LOOP
00076   C
00077   20      TYPE 104                                ASK FOR COMMAND
00078           ACCEPT 200,NCM                          GET IT
00079           IF (NCM .LE. 0 .OR. NCM .GT. 21) GO TO 20
00080           GO TO (1000,1100,1200,1300,1400,1500,1600,
00081           11700,1800,1900,2000,2100,2200,2300,2400,
00082           22500,2600,2700,2800,2900,3000)NCM
00083   C
00084   C       THIS IS THE 'COMMAND INTERPRETER' USING A SIMPLE
00085   C       NUMERICAL SWITCH TO SELECT A DESTINATION LABEL
00086   C
00087   C
00088   C       FORMAT BLOCK
00089   C
00090   100     FORMAT (///,T5,'THIS PROGRAM WILL HELP YOU SELECT THE
00091           1 BEST COMPONENTS.',/,T5,'SOME PARAMETERS ARE NEEDED.
00092           2 ANSWER THE FOLLOWING QUESTIONS.',/,T5,'FINISHING BY
00093           3 CR.',//,T10,'ENTER # OF CLASSES : ',$)
00094   101     FORMAT (T10,'ENTER TOTAL # OF COMPONENTS : ',$)
00095   102     FORMAT (T10,'ENTER FILENAME FOR CLASS #',I3,' : ',$)
00096   103     FORMAT (/,T5,'THE INITIALIZATION IS COMPLETE. YOU NOW
00097           1 ENTER THE COMMAND LOOP.',/,T5,'THE 21 COMMANDS ARE
00098           2 LISTED IN THE INSTRUCTION NOTICE.',///)
00099   104     FORMAT (/,T10,'TYPE IN YOUR COMMAND CODE (1-21) : ',$)
00100   118     FORMAT (T10,'ENTER OUTPUT FILENAME FOR THIS CLASS : ',$)
00101   C
00102   200     FORMAT (I3)
00103   201     FORMAT (A10)
00104   202     FORMAT (D22.15)
00105   C
00106   C       END OF FORMAT BLOCK
00107   C
00108   C
00109   C       FIRST COMMAND
00110   C
00111   1000    PRINT 139                               PRINT A LINE OF SEPARATION
00112           GO TO 20                                NEXT COMMAND
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-2


00113   C
00114   C       SECOND COMMAND
00115   C
00116   1100    DO 1110 I=1,NCO
00117   1110    IND(I) = 0                              DELETE ALL COMPONENTS
00118           DO 1120 I=1,NDIS
00119           WEIGHT(I) = 1.                          RESET WEIGHTS TO 1.
00120   1120    DIST(I) = 0.
00121           DO 1130 K=1,NCL
00122           DO 1130 I=1,NCO
00123           DO 1130 J=1,NCO
00124   1130    C2(I,J,K) = 0.                          RESET ALL CURRENT INVERSES
00125           GO TO 20                                NEXT COMMAND
00126   C
00127   C       THIRD COMMAND
00128   C
00129   1200    TYPE 110
00130           ACCEPT 200,I                            GET COMPONENT SPECIFIC.
00131           IF (I .GT. NCO .OR. I .LE. 0) GO TO 1200
00132           IF (IND(I) .NE. 0) GO TO 1210
00133   C
00134   C       THE INPUT HAS NOW BEEN THOROUGHLY CHECKED
00135   C
00136           IND(I) = I                              ADD SPECIFIED COMP.
00137           CALL DISTUP(I,NCO,NCL)                  UPDATE DISTANCES
00138           CALL INVUP(I,NCO,NCL)                   UPDATE INVERSES
00139           GO TO 20                                NEXT COMMAND
00140   C
00141   1210    TYPE 132                                COMP. ALREADY SELECTED
00142           GO TO 20
00143   C
00144   C       FOURTH COMMAND
00145   C
00146   1300    TYPE 111
00147           ACCEPT 200,I                            GET COMPONENT SPECIFIC.
00148           IF (I .GT. NCO .OR. I .LE. 0) GO TO 1300
00149           IF (IND(I) .EQ. 0) GO TO 1310
00150   C
00151   C       CHECK FINISHED
00152   C
00153           IND(I) = 0                              DELETE SPECIF. COMPON.
00154           CALL DISTDN(I,NCO,NCL)                  UPDATE DISTANCES
00155           CALL INVDN(I,NCO,NCL)                   UPDATE INVERSES
00156           GO TO 20                                NEXT COMMAND
00157   C
00158   1310    TYPE 133                                COMP. ALREADY DELETED
00159           GO TO 20
00160   C
00161   C       FIFTH COMMAND
00162   C
00163   1400    TYPE 117                                ASK FOR # OF STEPS
00164           ACCEPT 200,NST
00165           IF (NST .LE. 0 .OR. NST .GT. NCO) GO TO 1400
00166           DO 1460 MM=1,NST                        LOOP THROUGH # OF STEPS
00167           DO 1410 I=1,NCOMP
00168   1410    CONT(I) = 0.                            CLEAR CONTRIB.
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-3


00169           WT = 0.                                 TO AVERAGE CONTR.
00170           DO 1450 K=1,NCL
00171           DO 1450 L=1,NCL
00172           IF (K .EQ. L) GO TO 1450
00173           CALL UPDAT1(NCO,NCL,K,L,WT)             UPDATE COMP. CONTRIB.
%FTNDIM LINE:00173 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00173 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00174   1450    CONTINUE                                NEXT CLASS
00175   C
00176   C       THE CONTRIBUTION OF EACH UNSELECTED COMPONENT TO
00177   C       THE AVERAGE INTERCLASS DISTANCE HAS NOW BEEN COMPUTED
00178   C       THE LARGEST 'CONTRIBUTOR' WILL BE SELECTED
00179   C
00180           CALL MAX(I,NCO)                         RETURNS INDEX OF LARGEST IN I
00181           IF (I .EQ. 0) GO TO 1480                ALL ALREADY SELECTED
00182           CONTR = CONT(I)/WT
00183           IF (CONTR .LT. TA) GO TO 1470           BELOW THRESHOL-TO-SELECT
00184           IND(I) = I                              SELECT COMPONENT
00185           TYPE 120,I,CONTR
00186           PRINT 120,I,CONTR                       INFORMATION OUTPUT
00187           CALL DISTUP(I,NCO,NCL)                  UPDATE  DISTANCES
00188           CALL INVUP(I,NCO,NCL)                   UPDATE INVERSES
00189   1460    CONTINUE                                NEXT STEP
00190           GO TO 20                                NEXT COMMAND
00191   C
00192   1470    TYPE 136
00193           PRINT 136                               BELOW THRESHOL-TO-SELECT
00194           GO TO 20
00195   C
00196   1480    TYPE 134                                ALL ALREADY SELECTED
00197           PRINT 134
00198           GO TO 20
00199   C
00200   C       SIXTH COMMAND
00201   C
00202   1500    TYPE 117                                ASK FOR # OF STEPS
00203           ACCEPT 200,NST
00204           IF (NST .LE. 0 .OR. NST .GT. NCO) GO TO 1500
00205           DO 1560 MM=1,NST                        LOOP THROUGH ALL STEPS
00206           DO 1510 I=1,NCOMP
00207   1510    CONT(I) = 0.                            RESET CONTRIB.
00208           WT = 0.
00209           DO 1550 K=1,NCL
00210           DO 1550 L=1,NCL
00211           IF (K .EQ. L) GO TO 1550
00212           CALL UPDAT2(NCO,NCL,K,L,WT)             UPDATE COMP. CONTRIB.
%FTNDIM LINE:00212 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00212 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00213   1550    CONTINUE
00214   C
00215   C       THE CONTRIBUTION OF EACH SELECTED COMPONENT TO
00216   C       THE AVERAGE INTERCLASS DISTANCE HAS NOW BEEN COMPUTED
00217   C       THE SMALLEST 'CONTRIBUTOR' WILL BE REJECTED
00218   C
00219           CALL MIN(I,NCO)                         THE INDEX OF SMALLEST IN I
00220           IF (I .EQ. 0) GO TO 1580                ALL ALREADY REJECTED
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-4


00221           CONTR = CONT(I)/WT
00222           IF (CONTR .GT. TR) GO TO 1570     ABOVE THRESHOLD-TO-REJECT
00223           IND(I) = 0                        REJECT COMPONENT
00224           TYPE 121,I,CONTR
00225           PRINT 121,I,CONTR                 INFORMATION OUTPUT
00226           CALL DISTDN(I,NCO,NCL)            UPDATE DISTANCES
00227           CALL INVDN(I,NCO,NCL)             UPDATE INVERSES
00228   1560    CONTINUE                          NEXT STEP
00229           GO TO 20                          NEXT COMMAND
00230   C
00231   1570    TYPE 137                          ABOVE THRESHOLD-TO-REJECT
00232           PRINT 137
00233           GO TO 20
00234   C
00235   1580    TYPE 135
00236           PRINT 135                         ALL COMP. ALREADY REJECTED
00237           GO TO 20
00238   C
00239   C       SEVENTH COMMAND
00240   C
00241   1600    TYPE 117                          ASK FOR # OF STEPS
00242           ACCEPT 200,NST
00243           IF (NST .LE. 0 .OR. NST .GT. NCO) GO TO 1600
00244   1605    TYPE 112                          ASK FOR CLASS #
00245           ACCEPT 200,K
00246           IF (K .LE. 0 .OR. K .GT. NCL) GO TO 1605
00247           DO 1660 MM=1,NST                  LOOP THROUGH ALL STEPS
00248           DO 1610 I=1,NCOMP
00249   1610    CONT(I) = 0.                      RESET CONTRIB.
00250           WT = 0.
00251           DO 1650 L=1,NCL                   LOOP TO COMPUTE DIST.
00252           IF (L .EQ. K) GO TO 1650          SKIP CLASS ITSELF
00253           CALL UPDAT1(NCO,NCL,K,L,WT)       UPDATE COMP. CONTRIB.
%FTNDIM LINE:00253 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00254   1650    CONTINUE                          NEXT CLASS
00255   C
00256   C       THIS LOOP HAS COMPUTED THE CONTRIBUTION OF EACH
00257   C       UNSELECTED COMPONENT TO THE TOTAL INTERCLASS DISTANCE
00258   C       NOW THE LARGEST 'CONTRIBUTOR' WILL BE SELECTED
00259   C
00260           CALL MAX(I,NCO)                   RETURNS THE INDEX IN I
00261           IF (I .EQ. 0) GO TO 1480          ALL COMP. ALREADY SEL.
00262           CONTR = CONT(I)/WT                AVERAGE CONTRIBUTION
00263           IF (CONTR .LT. TA) GO TO 1470     IS IT BELOW THRESHOLD ?
00264           IND(I) = I                        NO, SO ADD COMPONENT
00265           TYPE 120,I,CONTR                  OUTPUT DATA
00266           PRINT 120,I,CONTR
00267           CALL DISTUP(I,NCO,NCL)
00268           CALL INVUP(I,NCO,NCL)             UPDATE INVERSES
00269   1660    CONTINUE                          NEXT STEP
00270           GO TO 20                          NEXT COMMAND
00271   C
00272   C       EIGHTH COMMAND
00273   C
00274   1700    TYPE 117                          ASK FOR # OF STEPS
00275           ACCEPT 200,NST
```

```
SELEC2  SELEC2.FOR      FORTRAN V.S(515) /KI    2-AUG-77        14:43   PAGE 1-5


00276           IF (NST .LE. 0 .OR. NST .GT. NCO) GO TO 1700
00277   1705    TYPE 112                                ASK FOR CLASS #
00278           ACCEPT 200,K
00279           IF (K .LE. 0 .OR. K .GT. NCL) GO TO 1705
00280           DO 1760 MM=1,NST                        LOOP THROUGH ALL STEPS
00281           DO 1710 I=1,NCOMP
00282   1710    CONT(I) = 0.                            RESET CONTRIB.
00283           WT = 0.
00284           DO 1750 L=1,NCL
00285           IF (L .EQ. K) GO TO 1750                SKIP CLASS ITSELF
00286           CALL UPDAT2(NCO,NCL,K,L,WT)             UPDATES COMP. CONTRIB.
%FTNDIM LINE:00286 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00287   1750    CONTINUE                                NEXT CLASS
00288   C
00289   C       THIS LOOP HAS COMPUTED THE CONTRIBUTION OF EACH SELECTED
00290   C       COMPONENT TO THE TOTAL INTERCLASS DISTANCE. THE SMALLEST
00291   C       'CONTRIBUTOR' WILL NOW BE DETERMINED AND REJECTED
00292   C
00293           CALL MIN(I,NCO)                         RETURNS INDEX IN I
00294           IF (I .EQ. 0) GO TO 1580                ERROR
00295           CONTR = CONT(I)/WT                      AVERAGE CONTRIBUTION
00296           IF (CONTR .GT. TR) GO TO 1570           IS IT ABOVE THRESHOLD ?
00297           IND(I) = 0                              NO, SO REJECT COMP.
00298           TYPE 121,I,CONTR                        DISPLAY INFORMATION
00299           PRINT 121,I,CONTR
00300           CALL DISTON(I,NCO,NCL)                  UPDATE DISTANCES
00301           CALL INVON(I,NCO,NCL)                   UPDATE INVERSES
00302   1760    CONTINUE                                NEXT STEP
00303           GO TO 20                                NEXT COMMAND
00304   C
00305   C
00306   C       NINTH COMMAND
00307   C
00308   1800    TYPE 117                                ASK FOR # OF STEPS
00309           ACCEPT 200,NST
00310           IF (NST .LE. 0 .OR. NST .GT. NCO) GO TO 1800
00311   1805    TYPE 113
00312           ACCEPT 203,K,L                          GET CLASS #S
00313           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00314          1 K .LE. 0 .OR. L .LE. 0) GO TO 1805
00315           IF (K .EQ. L) GO TO 2060
00316           DO 1860 MM=1,NST                        LOOP THROUGH ALL STEPS
00317           DO 1810 I=1,NCOMP
00318   1810    CONT(I) = 0.                            RESET CONTR.
00319           WT = 0.
00320           CALL UPDAT1(NCO,NCL,K,L,WT)             UPDATE COMP. CONTRIB.
00321           CALL MAX(I,NCO)
00322           IF (I .EQ. 0) GO TO 1480
00323           CONTR = CONT(I)/WT
00324           IF (CONTR .LE. TA) GO TO 1470           BELOW THRESHOLD-TO-SELECT
00325           IND(I) = I                              SELECT COMPONENT
00326           TYPE 120,I,CONTR
00327           PRINT 120,I,CONTR
00328           CALL DISTUP(I,NCO,NCL)
00329           CALL INVUP(I,NCO,NCL)                   UPDATE INVERSES
00330   1860    CONTINUE                                NEXT STEP
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-6

00331           GO TO 20                                NEXT COMMAND
00332   C
00333   C       TENTH COMMAND
00334   C
00335   1900    TYPE 117                                ASK FOR NUMBER OF STEPS
00336           ACCEPT 200,NST
00337           IF (NST .LE. 0 .OR. NST .GT. NCU) GO TO 1900
00338   1905    TYPE 113
00339           ACCEPT 203,K,L                          GET CLASS #S
00340           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00341           2 K .LE. 0 .OR. L .LE. 0) GO TO 1905
00342           IF (K .EQ. L) GO TO 2060
00343           DO 1960 MM=1,NST                        LOOP THROUGH ALL STEPS
00344           DO 1910 I=1,NCOMP
00345   1910    CONT(I) = 0.                            RESET CONTR.
00346           WT = 0.
00347           CALL UPDAT2(NCO,NCL,K,L,WT)             UPDATE COMP. CONTRIB.
00348           CALL MIN(I,NCO)
00349           IF (I .EQ. 0) GO TO 1580                ALL REJECTED
00350           CONTR = CONT(I)/WT
00351           IF (CONTR .GT. TR) GO TO 1570           ABOVE THRESHOLD-TO-REJECT
00352           IND(I) = 0                              REJECT COMPONENT
00353           TYPE 121,I,CONTR
00354           PRINT 121,I,CONTR
00355           CALL DISTDN(I,NCO,NCL)
00356           CALL INVDN(I,NCO,NCL)
00357   1960    CONTINUE                                NEXT STEP
00358           GO TO 20                                NEXT COMMAND
00359   C
00360   C       ELEVENTH COMMAND
00361   C
00362   2000    TYPE 113
00363           ACCEPT 203,K,L                          GET CLASS #S
00364           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00365           1 K .LE. 0 .OR. L .LE. 0) GO TO 2000
00366           IF (K .EQ. L) GO TO 2060
00367           NORD1 = (NCL-1)*(K-1)+L-1
00368           NORD2 = (NCL-1)*(L-1)+K-1               INDEX #S IN DIST
00369           IF (K .GT. L) NORD1=NORD1+1
00370           IF (K .LT. L) NORD2=NORD2+1
00371           HMS = 0.
00372           IF (DIST(NORD1) .EQ. 0. .AND. DIST(NORD2) .EQ. 0.) GO TO 2010
00373           HMS = 4.*DIST(NORD1)*DIST(NORD2)/
00374           1((SQRT(DIST(NORD1))+SQRT(DIST(NORD2)))**2)
00375   C
00376   C       NORD1 & 2 ARE THE DISPLACEMENTS IN THE ARRAY C (DIST.)
00377   C       HMS STANDS FOR HARMONIC MEAN SQUARE
00378   C
00379   2010    TYPE 122,K,L,K,DIST(NORD1),L,DIST(NORD2),HMS
00380           GO TO 20                                NEXT COMMAND
00381   C
00382   2060    TYPE 123                                ERROR MESSAGE
00383           GO TO 20
00384   C
00385   C       TWELFTH COMMAND
00386   C
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-7


00387   2100    TYPE 124
00388           CALL PACK(NCO,J)
00389           TYPE 125,(CONT(I),I=1,J)          TYPE SELECTED COMP. #S
00390           TYPE 130
00391           GO TO 20                          NEXT COMMAND
00392   C
00393   C       THIRTEENTH COMMAND
00394   C
00395   2200    CALL PARAM(NCO,NCL,NDIS,NSB,VMIN,AVER,SD,I,J)
00396           TYPE 126,NSB,AVER,SD,VMIN,I,J,TA,TR      SHORT STATUS
00397           GO TO 20                          NEXT COMMAND
00398   C
00399   C       FOURTEENTH COMMAND
00400   C
00401   2300    CALL PARAM(NCO,NCL,NDIS,NSB,VMIN,AVER,SD,I,J)
00402           PRINT 126,NSB,AVER,SD,VMIN,I,J,TA,TR     SHORT STATUS
00403           PRINT 127
00404           CALL PACK(NCO,J)
00405           PRINT 125,(CONT(I),I=1,J)         LIST OF SEL. COMP.
00406           PRINT 130
00407           GO TO 20                          NEXT COMMAND
00408   C
00409   C       FIFTEENTH COMMAND
00410   C
00411   2400    CALL PARAM(NCO,NCL,NDIS,NSB,VMIN,AVER,SD,I,J)
00412           PRINT 126,NSB,AVER,SD,VMIN,I,J,TA,TR     SHORT STATUS
00413           PRINT 127
00414           CALL PACK(NCO,J)
00415           PRINT 125,(CONT(I),I=1,J)         LIST OF SEL. COMP.
00416           PRINT 130
00417           PRINT 128
00418           K = 0
00419   C
00420   C       THE PROGRAM WILL NOW PRINT ALL INTERCLASS DISTANCES,
00421   C       ORDERED BY CLASS
00422   C
00423           DO 2410 I=1,NCL
00424           DO 2410 J=1,NCL
00425           IF (I .EQ. J) GO TO 2410          NOT A POSSIBLE DISTANCE
00426           NORD1 = (NCL-1)*(I-1)+J-1
00427           IF (I .GT. J) NORD1=NORD1+1
00428           IF (K .NE. 0) GO TO 2405          1ST OR 2ND PRINT COLUMN
00429           K = 1
00430           PRINT 131                         1ST COLUMN
00431           PRINT 129,I,J,DIST(NORD1),WEIGHT(NORD1)
00432           GO TO 2410
00433   C
00434   2405    K = 0                             2ND COLUMN
00435           PRINT 129,I,J,DIST(NORD1),WEIGHT(NORD1)
00436   C
00437   2410    CONTINUE
00438           PRINT 130
00439           GO TO 20                          NEXT COMMAND
00440   C
00441   C       SIXTEENTH COMMAND
00442   C
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-8

00443   2500    TYPE 114                                ASK FOR THRESHOLD-TO-ADD
00444           ACCEPT 204,TA
00445           GO TO 20
00446   C
00447   C       SEVENTEENTH COMMAND
00448   C
00449   2600    TYPE 115                                ASK FOR THRESHOLD-TO-REJECT
00450           ACCEPT 204,TR
00451           GO TO 20
00452   C
00453   C       EIGHTEENTH COMMAND
00454   C
00455   2700    TYPE 116                                ASK FOR CLASS #S,WEIGHT
00456           ACCEPT 205,I,J,WGHT
00457           IF (I .LE. 0 .OR. I .GT. NCL .OR.
00458           1 J .LE. 0 .OR. J .GT. NCL) GO TO 2700
00459           IF (I .EQ. J) GO TO 2060
00460           NORD1 = (NCL-1)*(I-1)+J-1
00461           IF (I .GT. J) NORD1=NORD1+1
00462           WEIGHT(NORD1) = WGHT
00463           GO TO 20
00464   C
00465   C       NINETEENTH COMMAND
00466   C
00467   2800    TYPE 113                                ASK FOR CLASS #S
00468           ACCEPT 203,I,J
00469           IF (I .LE. 0 .OR. I .GT. NCL .OR.
00470           1 J .LE. 0 .OR. J .GT. NCL) GO TO 2800
00471           IF (I .EQ. J) GO TO 2060
00472           NORD1 = (NCL-1)*(I-1)+J-1
00473           IF (I .GT. J) NORD1=NORD1+1
00474           TYPE 138,WEIGHT(NORD1)                  TYPE WEIGHT FOR THESE CLASSES
00475           GO TO 20
00476   C
00477   C       TWENTIETH COMMAND
00478   C
00479   2900    CALL PARAM(NCO,NCL,NDIS,NSB,VMIN,AVER,SD,I,J)
00480           TYPE 140,VMIN,I,J                       MINIMUM DISTANCE
00481           GO TO 20
00482   C
00483   C       TWENTY-FIRST COMMAND
00484   C
00485   3000    DO 3010 I=1,NCL                         LOOP TO WRITE INVERSE MATRICES
00486           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D2(I))
00487           DO 3003 J=1,NCO
00488           IF (IND(J) .EQ. 0) GO TO 3003           ONLY SEL. COMP. ARE WRITTEN
00489           WRITE (20,206),M(J,I)                   WRITE MEAN VECTOR
00490   3003    CONTINUE
00491           DO 3006 J=1,NCO
00492           IF (IND(J) .EQ. 0) GO TO 3006
00493           DO 3006 K=1,NCO
00494           IF (IND(K) .EQ. 0) GO TO 3006           ONLY SEL. COMP. ARE WRITTEN
00495           WRITE (20,206),C2(J,K,I)                WRITE INVERSE MATRIX
00496   3006    CONTINUE
00497   C
00498   C       THE I-TH CLASS' MEAN VECTOR AND INVERSE COV. MATRIX
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-9


00499   C       HAVE NOW BEEN WRITTEN TO DISK
00500   C
00501           CLOSE (UNIT=20,DEVICE='DSK',FILE=D2(I))
00502   3010    CONTINUE                                NEXT CLASS
00503   C
00504   C       ALL INVERSE COV. MATRICES HAVE BEEN SAVED ON DISK
00505   C
00506           STOP                                    END OF PROGRAM
00507   C
00508   C
00509   C       FORMAT BLOCK
00510   C
00511   110     FORMAT (T10,'TYPE IN THE # OF THE COMP. TO BE ADDED : ',$)
00512   111     FORMAT (T10,'TYPE IN THE # OF THE COMP. TO BE DELETED : ',$)
00513   112     FORMAT (T10,'TYPE IN CLASS # : ',$)
00514   113     FORMAT (T10,'TYPE IN BOTH CLASS NUMBERS, SEP. BY A COMMA : ',$)
00515   114     FORMAT (T10,'TYPE IN THRESHOLD-TO-SELECT : ',$)
00516   115     FORMAT (T10,'TYPE IN THRESHOLD-TO-REJECT : ',$)
00517   116     FORMAT (T10,'TYPE IN BOTH CLASS #S & WEIGHT, SEP.
00518           1 BY A COMMA : ',$)
00519   117     FORMAT (T10,'TYPE IN # OF STEPS : ',$)
00520   C
00521   120     FORMAT (T10,'THE SELECTED COMP. IS #',I3,', WITH A
00522           1 CONTR. OF ',E11.4,/)
00523   121     FORMAT (T10,'THE REJECTED COMP. IS #',I3,', WHICH HAD A
00524           1 CONTR. OF ',E11.4,/)
00525   122     FORMAT (T10,'THE DISTANCES BETWEEN CLASS',I3,' AND CLASS'
00526           1,I3,' ARE :',/,T10,'--REL. TO CLASS',I3,' : ',E11.4,/,
00527           2T10,'--REL. TO CLASS',I3,' : ',E11.4,/,T10,
00528           3'--HARMONIC MEAN SQUARE : ',E11.4,/)
00529   123     FORMAT (T10,'CLASS #S MUST BE DIFFERENT  ',/)
00530   124     FORMAT (T10,'THE FOLLOWING COMPONENTS ARE SELECTED :')
00531   125     FORMAT (/,10(3X,F4.0))
00532   126     FORMAT (T10,'THE PRESENT STATUS IS :',/,T10,
00533           1'--NUMBER OF SELECTED COMPONENTS :',I4,/,T10,
00534           2'--AVERAGE INTERCLASS DISTANCE : ',E11.4,/,T10,
00535           3'  WITH A  STANDARD DEV. OF    : ',E11.4,/,T10,
00536           4'--THE MINIMUM RMS DISTANCE IS : ',E11.4,/,T10,
00537           5'  BETWEEN CLASSES',I3,' AND',I3,/,T10,
00538           6'--THE THRESHOLD-TO-ADD IS      : ',E11.4,/,T10,
00539           7'  THE THRESHOLD-TO-REJECT IS  : ',E11.4,/)
00540   127     FORMAT (T10,'THE SELECTED COMPONENTS ARE :')
00541   128     FORMAT (/,T10,'THE INTERCLASS DISTANCES ARE :',//,T10,
00542           12('FROM:  TO:  DISTANCE:   WEIGHT:     '))
00543   129     FORMAT (4X,I3,3X,I3,2X,E11.4,2X,F6.3,3X,$)
00544   130     FORMAT (//)
00545   131     FORMAT (/,T7,$)
00546   132     FORMAT (/,T10,'THIS COMPONENT IS ALREADY SELECTED  ',/)
00547   133     FORMAT (/,T10,'THIS COMPONENT IS ALREADY REJECTED  ',/)
00548   134     FORMAT (/,T10,'ERROR : ALL COMPONENTS ALREADY SELECTED
00549           1  ',/)
00550   135     FORMAT (/,T10,'ERROR : ALL COMPONENTS ALREADY REJECTED
00551           1  ',/)
00552   136     FORMAT (/,T10,'CONTRIB. BELOW THRESHOLD-TO-ADD  ',/)
00553   137     FORMAT (/,T10,'CONTRIB. ABOVE THRESHOLD-TO-REJECT  ',/)
00554   138     FORMAT (/,T10,'THE WEIGHT IS : ',F6.3,/)
```

```
SELEC2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-10


00555   139     FORMAT (/,72('*'),/)
00556   140     FORMAT (T10,'THE MINIMUM HMS DISTANCE IS : ',E11.4,/,
00557           1T10,'BETWEEN CLASSES',I3,' AND',I3,/)
00558   C
00559   203     FORMAT (2I)
00560   204     FORMAT (F10.5)
00561   205     FORMAT (2I,F6.3)
00562   206     FORMAT (E14.7)
00563   C
00564   C       END OF FORMAT BLOCK
00565   C       AND END OF PROGRAM
00566           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED

MIN     SQRT.   DISTUP  MAX     DISTDN  INVDN
INVUP   UPDAT1  UPDAT2  PACK    PARAM

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 .S0020 1       *NCO    2       *NORD1  3       *SD     4       .S0037 5
 .S0036 6        .S0035 7        .S0034 10       .S0033 11       .S0032 12
*K      13       .S0031 14       .S0030 15      *NCL    16      *TR     17
*TA     20       .S0044 21       .S0043 22       .S0042 23       .S0041 24
 D2     25       .S0040 45      *WGHT   46      *HMS    47      *NDIS   50
*NCM    51      *MM     52      *J      53       .S0007 54       .S0006 55
 .S0005 56       D      57      *NSE    77       .S0004 100     *NST    101
 .S0003 102      .S0002 103      .S0001 104      .S0000 105     *AVER   106
 .S0017 107      .S0016 110      .S0015 111      .S0014 112      .S0013 113
 .S0012 114      .S0011 115      .S0010 116     *L      117     *NORD2  120
*WT     121     *I      122     *VMIN   123      .S0027 124      .S0026 125
 .S0025 126      .S0024 127      .S0023 130      .S0022 131      .S0021 132
*CONTR  133

TEMPORARIES

 .Q0000 1061     .Q0001 1062

%FTNWRN   SELEC2        NO FATAL ERRORS AND 6 WARNINGS
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MAX(I,NCO)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z
00012           I = 0
00013           VMAX = -1.E26
00014           DO 10   J=1,NCO
00015           IF (IND(J) .NE. 0) GO TO 10             ONLY UNSEL. COMP.
00016           IF (VMAX .GE. CONT(J)) GO TO 10         TEST FOR MAX.
00017           I = J
00018           VMAX = CONT(J)                          FOUND A LARGER
00019   10      CONTINUE
00020           RETURN
00021           END

COMMON BLOCKS

/.COMM./(*105444)
C1      *0      C2      *42020  IND     *104040 M       *104101 DIST    *105121
CONT    *105211 WEIGHT  *105252 Z       *105342

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *VMAX   2       *J      3       .S0000 4        *I      5

TEMPORARIES


MAX       NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MIN(I,NCO)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z
00012           I = 0
00013           VMIN = 1.E+26
00014           DO 10   J=1,NCO
00015           IF (IND(J) .EQ. 0) GO TO 10             ONLY SEL. COMP.
00016           IF (VMIN .LE. CONT(J)) GO TO 10         TEST FOR MIN.
00017           I = J
00018           VMIN = CONT(J)                          FOUND A LARGER
00019   10      CONTINUE
00020           RETURN
00021           END

COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *J      2       .50000 3        *I      4       *VMIN   5

TEMPORARIES


MIN       NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE PARAM(NCO,NCL,NDIS,NSB,VMIN,AVER,SD,L,MM)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z
00012   C
00013   C       THIS ROUTINE WILL COUNT THE NUMBER OF SEL. COMPON.
00014   C       AND COMPUTE THE AVERAGE INTERCLASS DISTANCE, ITS
00015   C       STANDARD DEVIATION AND ITS MINIMUM PART.
00016   C
00017           NSB = 0
00018           DO 10 K=1,NCO
00019           IF (IND(K) .NE. 0) NSB=NSB+1
00020   10      CONTINUE
00021           AVER = 0.
00022           L = 1
00023           MM = 2
00024           VMIN = 1.E26
00025           DO 20 K=1,NDIS
00026           AVER = AVER+DIST(K)/(NCL*(NCL-1))
00027           I = (K-1)/(NCL-1)+1
00028           J = K-(NCL-1)*(I-1)
00029           IF (J .GE. I) J=J+1
00030           NORD2 = (NCL-1)*(J-1)+I-1
00031           IF (J .GT. I) NORD2=NORD2+1
00032           HMS = 0.
00033           IF (DIST(K) .EQ. 0. .AND. DIST(NORD2) .EQ. 0.) GO TO 15
00034           HMS = 4.*DIST(K)*DIST(NORD2)/
00035          1((SQRT(DIST(K))+SQRT(DIST(NORD2)))**2)
00036   15      IF (VMIN .LE. HMS) GO TO 20
00037           VMIN = HMS
00038           L = I
00039           MM = J
00040   20      CONTINUE
00041           SD = 0.
00042           DO 30 I=1,NDIS
00043   30      SD = SD+((DIST(I)-AVER)**2)/(NCL*(NCL-1))
00044           SD = SQRT(SD)
00045           RETURN
00046           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342
```

```
PARAM   SELEC2.FOR      FORTRAN V.S(515) /KI    2-AUG-77        14:43   PAGE 1-1


SUBPROGRAMS CALLED

SQRT.

SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1       *SD      2       *K       3       *NCL     4       *MMS     5
*NDIS    6       *J       7       *MM      10      *NSB     11       .S0002 12
 .S0001 13        .S0000 14       *AVER    15      *L       16      *NORD2  17
*I       20      *VMIN    21

TEMPORARIES

 .A0016 22        .Q0000 23        .Q0001 24

PARAM      NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE UPDAT2(NCO,NCL,K,L,WT)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z,TEMP
00012   C
00013   C       THIS ROUTINE WILL COMPUTE THE CONTRIBUTION OF SELECTED
00014   C       COMPONENTS, ONE STEP AT A TIME (ADDING ONE CONTRIBUTION
00015   C       EVERY TIME IT'S CALLED)
00016   C
00017           NORD = (NCL-1)*(K-1)+L-1
00018           IF (K .GT. L) NORD=NORD+1
00019           WT = WT + WEIGHT(NORD)
00020           DO 100 N=1,NCO                          GENERAL LOOP FOR COMP.
00021   C
00022   C
00023   C       THIS LOOP WILL COMPUTE THE INDIVIDUAL CONTRIBUTION OF
00024   C       EACH SEL. COMPONENT
00025   C
00026           IF (IND(N) .EQ. 0) GO TO 100
00027   C       FROM THE SCALAR PRODUCT M(K)-M(L) * C2(N,K)
00028   C
00029           TEMP = 0.
00030           DO 40 I=1,NCO
00031           IF (IND(I) .EQ. 0) GO TO 40
00032           IF (I .EQ. N) GO TO 40                  SKIP THIS INDEX
00033           TEMP = TEMP+(M(I,K)-M(I,L))*C2(I,N,K)
00034   40      CONTINUE
00035           TEMP = (TEMP+C2(N,N,K)*(M(N,K)-M(N,L)))**2
00036           CONT(N) = CONT(N)+TEMP*WEIGHT(NORD)/C2(N,N,K)
00037   C
00038   C       THE CONTRIBUTION OF THE N-TH COMPONENT IS NOW COMPUTED
00039   C
00040   100     CONTINUE
00041           RETURN
00042           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED
```

```
UPDAT2  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1


SCALARS AND ARRAYS   "#" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *N      2       *K      3       *NCL    4       *NORD   5
 .S0001 6        .S0000 7        TEMP   10       *L     12       *I     13
*%T     14

TEMPORARIES


UPDAT2    NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE UPDAT1(NCO,NCL,K,L,WT)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z,BETA,TEMP
00012   C
00013   C       THIS ROUTINE WILL COMPUTE THE CONTRIBUTION OF UNSELECTED
00014   C       COMPONENTS, ONE STEP AT A TIME (ADDING ONE CONTRIBUTION
00015   C       EVERY TIME IT'S CALLED)
00016   C
00017           NORD = (NCL-1)*(K-1)+L-1
00018           IF (K .GT. L) NORD=NORD+1
00019           WT = WT + WEIGHT(NORD)
00020           DO 100 N=1,NCO                          GENERAL LOOP FOR COMP.
00021   C
00022   C       THIS LOOP COMPUTES THE INDIVIDUAL CONTR. OF  EACH UNSEL.
00023   C       COMP.
00024   C
00025           IF (IND(N) .NE. 0) GO TO 100
00026   C
00027   C       COMPUTE THE VECTOR PRODUCT C2(K)*C1(M,K)
00028   C
00029           DO 20 I=1,NCO
00030           Z(I) = 0.
00031           IF (IND(I) .EQ. 0) GO TO 20
00032           DO 20 J=1,NCO
00033           IF (IND(J) .EQ. 0) GO TO 20
00034           Z(I) = Z(I)+C2(I,J,K)*C1(J,N,K)
00035   20      CONTINUE
00036   C
00037   C       FORM THE SCALAR PRODUCT
00038   C
00039           TEMP = 0.
00040           DO 30 I=1,NCO
00041           IF (IND(I) .EQ.  0) GO TO 30
00042           TEMP = TEMP + Z(I)*C1(N,I,K)
00043   30      CONTINUE
00044           BETA = C1(N,N,K)-TEMP
00045   C
00046   C       TEST FOR POSITIVE-DEFINITENESS
00047   C
00048           IF (BETA .GT. 0.) GO TO 150       ERROR NOT POS. DEF.
00049           TYPE 200
00050   200     FORMAT (T10,'WARNING : NO MORE POS.-DEF.  ',/)
00051   C
00052   C       COMPUTE NOW THE CONTRIBUTION
00053   C
00054   C       FORM THE SCALAR PRODUCT M(K)-M(L) * Z
00055   C
00056   150     TEMP = 0.
```

```
UPDAT1  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1


00057           DO 40 I=1,NCO
00058           IF (IND(I) .EQ. 0) GO TO 40
00059           TEMP = TEMP + Z(I)*(M(I,K)-M(I,L))
00060   40      CONTINUE
00061           TEMP = (TEMP-(M(N,K)-M(N,L)))**2
00062   C
00063   C       NOW ADD THIS CONTR. TO THE RUNNING VALUE
00064   C
00065           CONT(N) = CONT(N) + WEIGHT(NORD)*TEMP/BETA
00066   100     CONTINUE                                NEXT COMPONENT
00067           RETURN
00068           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS    "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *N      2       BETA    3       *K      5       *NCL    6
*NORD   7       *J      10      .S0004  11       .S0003 12       .S0002 13
 .S0001 14       .S0000 15      TEMP    16      *L      20      *I      21
*WT     22

TEMPORARIES

 .A0016 33

UPDAT1     NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515). /KI    2-AUG-77        14:43   PAGE 1


00001    C
00002    C
00003    C
00004            SUBROUTINE PACK(NCO,J)
00005            PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006            COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007            COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008            COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009            REAL CONT,DIST,WEIGHT
00010            INTEGER IND
00011            DOUBLE PRECISION C1,C2,M,Z
00012    C
00013    C       THIS ROUTINE WILL 'COMPACTIFY' THE ARRAY CONT, I.E.
00014    C       WILL JUSTIFY IT TO THE BEGINNING
00015    C
00016            DO 5 I=1,NCOMP
00017    5       CONT(I) = 0.
00018            J = 0
00019            DO 10 I=1,NCO
00020            IF (IND(I) .EQ. 0) GO TO 10      SKIP UNSEL. COMP.
00021            J = J+1
00022            CONT(J) = I
00023    10      CONTINUE
00024            RETURN
00025            END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "+" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

+NCO    1       +J      2       .S0001 3        .S0000 4        +I      5


TEMPORARIES


PACK      NO ERRORS DETECTED
```

```
DISTUP  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001           SUBROUTINE DISTUP(K,NCO,NCL)
00002           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00003           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00004           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00005           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00006           REAL CONT,DIST,WEIGHT
00007           INTEGER IND
00008           DOUBLE PRECISION C1,C2,M,Z,BETA,TEMP
00009   C
00010           DO 300 II=1,NCL
00011           DO 300 JJ=1,NCL
00012           IF (II .EQ. JJ) GO TO 300          LOOP THROUGH ALL DISTANCES
00013           NORD = (NCL-1)*(II-1)+JJ-1
00014           IF (II .GT. JJ) NORD=NORD+1        INDEX IN DIST
00015   C
00016   C       COMPUTE THE VECTOR PRODUCT C2(II)*C1(II,K)
00017   C
00018           DO 20 I=1,NCO
00019           Z(I) = 0.
00020           IF (IND(I) .EQ. 0) GO TO 20        ONLY SEL. COMP.
00021           IF (I .EQ.  K) GO TO 20
00022           DO 20 J=1,NCO
00023           IF (IND(J) .EQ. 0) GO TO 20
00024           IF (J .EQ. K) GO TO 20
00025           Z(I) = Z(I)+C2(I,J,II)*C1(J,K,II)
00026   20      CONTINUE
00027   C
00028   C       FORM THE SCALAR PRODUCT Z*C1(K,II)
00029   C
00030           TEMP = 0.
00031           DO 30 I=1,NCO
00032           IF (IND(I) .EQ. 0) GO TO 30        ONLY SEL. COMP.
00033           IF (I .EQ. K) GO TO 30
00034           TEMP = TEMP+Z(I)*C1(I,K,II)
00035   30      CONTINUE
00036           BETA = C1(K,K,II)-TEMP
00037   C
00038   C       TEST FOR POSITIVE DEFINITENESS
00039   C
00040           IF (BETA .GT. 0.) GO TO 150        ERROR, NOT POS.DEF.
00041           TYPE 200                           SO TYPE A WARNING
00042   200     FORMAT (T10,'WARNING : NO MORE POS.DEF.',/)
00043   C
00044   C       FORM THE SCALAR PRODUCT M(L)*Z
00045   C
00046   150     TEMP = 0.
00047           DO 40 I=1,NCO
00048           IF (IND(I) .EQ. 0) GO TO 40        ONLY SEL. COMP.
00049           IF (I .EQ. K) GO TO 40
00050           TEMP = TEMP+Z(I)*(M(I,II)-M(I,JJ))
00051   40      CONTINUE
00052           TEMP = (TEMP-(M(K,II)-M(K,JJ)))**2
00053   C
00054   C       NOW ADD THIS CONTR. TO THE PRESENT VALUE
00055   C
00056           DIST(NORD) = DIST(NORD)+TEMP/BETA
```

```
DISTUP  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1


00057   300     CONTINUE                        NEXT CLASSES
00058   C
00059           RETURN
00060   C
00061           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS    "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *JJ     2       BETA    3       *K      5       *NCL    6
*NORD   7       *J      10      .S0005  11      .S0004  12      .S0003  13
*II     14      .S0002  15      .S0001  16      .S0000  17      TEMP    20
*I      22

TEMPORARIES

 .A0016 33

DISTUP    NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE DISTDN(K,NCO,NCL)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00007           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00008           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00009           REAL CONT,DIST,WEIGHT
00010           INTEGER IND
00011           DOUBLE PRECISION C1,C2,M,Z,TEMP
00012   C
00013           DO 300 II=1,NCL
00014           DO 300 JJ=1,NCL
00015           IF (II .EQ. JJ) GO TO 300          LOOP THROUGH ALL DIST.
00016           NORD = (NCL-1)*(II-1)+JJ-1
00017           IF (II .GT. JJ) NORD=NORD+1
00018   C
00019   C       FORM THE SCALAR PRODUCT M(II)-M(JJ)  * C2(K,II)
00020   C
00021           TEMP = 0.
00022           DO 40 I=1,NCO
00023           IF (IND(I) .EQ. 0) GO TO 40        ONLY SEL. COMP.
00024           IF (I .EQ. K) GO TO 40             SKIP THIS INDEX
00025           TEMP = TEMP+(M(I,II)-M(I,JJ))*C2(I,K,II)
00026   40      CONTINUE
00027           TEMP = (TEMP+C2(K,K,II)*(M(K,II)-M(K,JJ)))**2
00028   C
00029   C       NOW SUBTRACT THIS CONTR. FROM THE PRESENT VALUE
00030   C
00031           DIST(NORD) = DIST(NORD) - TEMP/C2(K,K,II)
00032   300     CONTINUE                           NEXT CLASSES
00033   C
00034   C       THE MEASURE IS NOW UPDATED
00035   C
00036           RETURN
00037           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *JJ     2       *K      3       *NCL    4       *NORD   5
*II     6        .S0002 7        .S0001 10       .S0000 11       TEMP    12
*I      14
```

```
DISTDN  SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1

TEMPORARIES

DISTDN    NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE INVUP(K,NCO,NCL)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006   C
00007   C
00008           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00009           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00010           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00011           REAL CONT,DIST,WEIGHT
00012           INTEGER IND
00013           DOUBLE PRECISION C1,C2,M,Z,BETA,TEMP
00014   C
00015           DO  300 II=1,NCL                    GO THROUGH ALL CLASSES
00016   C
00017   C       COMPUTE THE VECTOR PRODUCT C2(II)*C1(K,II)
00018   C
00019           DO 10 I=1,NCO
00020           Z(I) = 0.
00021           IF (IND(I) .EQ. 0) GO TO 10
00022           IF (I .EQ. K) GO TO 10
00023           DO 10 J=1,NCO
00024           IF (IND(J) .EQ. 0) GO TO 10
00025           IF (J .EQ. K) GO TO 10
00026           Z(I) = Z(I) +C2(I,J,II)*C1(J,K,II)
00027   10      CONTINUE
00028   C
00029   C       FORM THE SCALAR PRODUCT Z*C1(K,II)
00030   C
00031           TEMP = 0.
00032           DO 30 I=1,NCO
00033           IF (IND(I) .EQ. 0) GO TO 30
00034           IF (I .EQ. K) GO TO 30
00035           TEMP = TEMP+Z(I)*C1(I,K,II)
00036   30      CONTINUE
00037           BETA = C1(K,K,II)-TEMP
00038   C
00039   C       TEST FOR POSITIVE-DEFINITENESS
00040   C
00041           IF (BETA .GT. 0.) GO TO 150         ERROR, NOT POSITIVE-DEFINITE
00042           TYPE 200                            SO TYPE A WARNING
00043   200     FORMAT (T10,'WARNING : NO MORE POS.DEF.',/)
00044   C
00045   150     C2(K,K,II) = 1./BETA
00046           DO 40 I=1,NCO
00047           IF (IND(I) .EQ. 0) GO TO 40
00048           IF (I .EQ. K) GO TO 40
00049           C2(I,K,II) = -Z(I)/BETA
00050           C2(K,I,II) = C2(I,K,II)
00051   40      CONTINUE
00052   C
00053   C       THE NEW ROW AND THE NEW COLUMN  OF THE INVERSE HAVE
00054   C       BEEN UPDATED. THE OLD BODY WILL NOW BE UPDATED.
00055   C
00056           DO 50 I=1,NCO
```

```
INVUP   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1


00057           IF (IND(I) .EQ. 0) GO TO 50
00058           IF (I .EQ. K) GO TO 50
00059           DO 50 J=I,NCO
00060           IF (IND(J) .EQ. 0) GO TO 50
00061           IF (J .EQ. K) GO TO 50
00062           C2(I,J,II) = C2(I,J,II)+Z(I)*Z(J)/BETA
00063           C2(J,I,II) = C2(I,J,II)
00064   50      CONTINUE
00065   C
00066   C       THE II-TH INVERSE HAS NOW BEEN UPDATED
00067   C
00068   300     CONTINUE                                NEXT INVERSE
00069           RETURN
00070   C
00071           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "$" NOT REFERENCED

*NCO    1       BETA    2       *K      4       *NCL    5       *J      6
 .S0006 7        .S0005 10        .S0004 11       .S0003 12       *II     13
 .S0002 14       .S0001 15        .S0000 16       TEMP   17       *I      21


TEMPORARIES

 .A0016 32

INVUP     NO ERRORS DETECTED
```

```
MAIN.   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE INVDN(K,NCO,NCL)
00005           PARAMETER NCLASS=8,NCOMP=33,NDIST=56
00006   C
00007   C
00008           COMMON C1(1:NCOMP,1:NCOMP,1:NCLASS),C2(1:NCOMP,1:NCOMP,1:NCLASS)
00009           COMMON IND(1:NCOMP),M(1:NCOMP,1:NCLASS),DIST(1:NDIST)
00010           COMMON CONT(1:NCOMP),WEIGHT(1:NDIST),Z(1:NCOMP)
00011           REAL CONT,DIST,WEIGHT
00012           INTEGER IND
00013           DOUBLE PRECISION C1,C2,M,Z
00014   C
00015           DO 300 II=1,NCL             LOOP THROUGH ALL INVERSES
00016   C
00017           DO 20 I=1,NCO
00018           IF (IND(I) .EQ. 0) GO TO 20
00019           IF (I .EQ. K) GO TO 20
00020           DO 20 J=I,NCO
00021           IF (IND(J) .EQ. 0) GO TO 20
00022           IF (J .EQ. K) GO TO 20
00023           C2(I,J,II) = C2(I,J,II)-C2(I,K,II)*C2(J,K,II)/C2(K,K,II)
00024           C2(J,I,II) = C2(I,J,II)
00025   20      CONTINUE
00026   C
00027   C       NOW THE MAIN BODY OF THE INVERSE HAS BEEN UPDATED
00028   C       THE EXCESS ROW AND COLUMN WILL BE ZEROED
00029   C
00030           DO 30 I=1,NCO
00031           C2(I,K,II) = 0.
00032   30      C2(K,I,II) = 0.
00033   C
00034   C       THE II-TH INVERSE HAS NOW BEEN UPDATED
00035   C
00036   300     CONTINUE                    NEXT INVERSES
00037           RETURN
00038           END


COMMON BLOCKS

/.COMM./(+105444)
C1      +0      C2      +42020  IND     +104040 M       +104101 DIST    +105121
CONT    +105211 WEIGHT  +105252 Z       +105342


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *K      2       *NCL    3       *J      4       .S0003 5
*II     6       .S0002 7        .S0001 10       .S0000 11       *I      12
```

```
INVON   SELEC2.FOR      FORTRAN V.5(515) /KI    2-AUG-77        14:43   PAGE 1-1


TEMPORARIES

 .00000 13        .00001 14

INVON     NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C       THIS PROGRAM IS DESIGNED TO PROVIDE INTERACTIVE
00003   C       SELECTION OF FEATURES (= COMPONENTS) FOR A PATTERN
00004   C       RECOGNITION SYSTEM
00005   C       THE INPUT TO THE PROGRAM CONSISTS OF A SET OF FILES,
00006   C       EACH OF WHICH CONTAINS A MEAN VECTOR AND AN INVERSE
00007   C       COVARIANCE MATRIX (AS PRODUCED, TYPICALLY, BY THE
00008   C       ROUTINE INVERT.FOR). THE USER MUST SUPPLY THE NUMBER
00009   C       OF THESE FILES, THE DIMENSION OF THE MATRICES, AS WELL
00010   C       AS THE NAMES OF THE FILES.
00011   C       THEN ONE HAS A SET OF 20 COMMANDS AT HIS DISPOSITION,
00012   C       NUMBERED FROM 1 TO 20. A COMMAND IS SELECTED BY TYPING
00013   C       IN ITS CODE NUMBER. SOME COMMANDS REQUIRE ADDITIONAL
00014   C       INFORMATION. THAT WILL BE REQUESTED FROM THE USER AS
00015   C       NECESSARY.
00016   C
00017           PROGRAM SELEC3
00018           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00019   C
00020   C       THESE PARAMETERS DEFINE UPPER BOUNDS ON THE NUMBER
00021   C       OF COMPONENTS AND CLASSES, BUT DO NOT INFLUENCE THE
00022   C       STORAGE SPACE ALLOCATION
00023   C
00024           COMMON C(1:NDIM)
00025           COMMON IND(1:NCOMP)
00026           REAL C
00027           INTEGER IND
00028           DOUBLE PRECISION D(1:NCLASS)
00029           DATA IND/NCOMP*0/,C/NDIM*0./
00030   C
00031   C       THE ARRAY D CONTAINS THE NAME OF THE FILES
00032   C       THE ARRAY C IS ALLOCATED IN FUNCTION OF THE USER'S
00033   C       ANSWERS. IT WILL CONTAIN, IN THIS ORDER, THE MEAN AND
00034   C       INVERSE COVARIANCE MATRIX OF EACH CLASS, THE PRESENT
00035   C       VALUE OF THE INTERCLASS DISTANCES, AND THE CONTRIBUTION
00036   C       OF EACH COMPONENT TO THE INTERCLASS DISTANCES.
00037   C       THE MATRICES ARE STORED ROW BY ROW. THE DISTANCES ARE
00038   C       STORED BETWEEN INDICES NDI AND NDE, THE WEIGHTS
00039   C       BETWEEN NDE+1 AND NDE+ND AND THE CONTRIBUTIONS
00040   C       BETWEEN NDE+ND+1 AND NCE
00041   C       IND KEEPS TRACE OF THE SELECTED COMPONENTS.
00042   C       (A ZERO MEANS THAT THE COMPONENT IS NOT SELECTED,
00043   C       ANY OTHER VALUES MEAN THAT THE COMPONENT IS SELECTED)
00044   C
00045   1       TYPE 100
00046           ACCEPT 200,NCL                          GET NUMBER OF CLASSES
00047           IF (NCL .GT. NCLASS .OR. NCL .LE. 0) GO TO 1
00048   2       TYPE 101
00049           ACCEPT 200,NCO                          GET NUMBER OF COMPON.
00050           IF (NCO .GT. NCOMP .OR. NCO .LE. 0) GO TO 2
00051           NC = NCO*(NCO+1)
00052           ND = NCL*(NCL-1)
00053           NCE = NCL*NC+NCO+2*ND
00054           IF (NCE .GT. NDIM) GO TO 1
00055           DO 10 I=1,NCL
00056           TYPE 102,I
```

```
SELEC3  SELEC3.FOR      FORTRAN V.S(515) /KI    27-JUL-77       20:09   PAGE 1-1


00057           ACCEPT 201,D(I)                     GET FILENAME OF CLASS I
00058   10      CONTINUE
00059           DO 15 I=1,NCL
00060           OPEN (MODE='BINARY',UNIT=20,DEVICE='DSK',FILE=D(I))
00061           NS = (I-1)*NC+1
00062           READ (20,202) (C(J),J=NS,NS+NC-1)
00063   C       READ IN MEAN VECTOR
00064           MS = I*NC
00065           READ (20,202) (C(J),J=NS+NC,MS)
00066   C       READ IN INVERSE COV. MATRIX
00067   15      CONTINUE
00068   C
00069   C       THIS LOOP HAS READ THE MEAN VECTORS AND INVERSE
00070   C       COVARIANCE MATRICES IN THE ARRAY C
00071   C
00072           NDI = MS+1
00073           NDE = MS+ND
00074           TR = 0.
00075           TA = 0.                             INITIALIZE THRESH. TO 0
00076           DO 17 I=NDE+1,NDE+ND
00077   17      C(I) = 1.                           INITIALIZE ALL WEIGHTS TO 1
00078           TYPE 103                            END OF INITIALIZATION
00079   C
00080   C       THE INITIALIZATION IS NOW FINISHED, AND THE PROGRAM
00081   C       ENTERS THE COMMAND LOOP
00082   C
00083   20      TYPE 104                            ASK FOR COMMAND
00084           ACCEPT 200,NCM                      GET IT
00085           IF (NCM .LE. 0 .OR. NCM .GT. 29) GO TO 20
00086           GO TO (1000,1100,1200,1300,1400,1500,1600,
00087           11700,1800,1900,2000,2100,2200,2300,2400,
00088           22500,2600,2700,2800,2900)NCM
00089   C
00090   C       THIS IS THE 'COMMAND INTERPRETER' USING A SIMPLE
00091   C       NUMERICAL SWITCH TO SELECT A DESTINATION LABEL
00092   C
00093   C
00094   C       FORMAT BLOCK
00095   C
00096   100     FORMAT (///,T5,'THIS PROGRAM WILL HELP YOU SELECT THE
00097           1 BEST COMPONENTS.',/,T5,'SOME PARAMETERS ARE NEEDED.
00098           2 ANSWER THE FOLLOWING QUESTIONS.',/,T5,'FINISHING BY
00099           3 CR.',///,T10,'ENTER # OF CLASSES : ',$)
00100   101     FORMAT (T10,'ENTER TOTAL # OF COMPONENTS : ',$)
00101   102     FORMAT (T10,'ENTER FILENAME FOR CLASS #',I3,' : ',$)
00102   103     FORMAT (/,T5,'THE INITIALIZATION IS COMPLETE. YOU NOW
00103           1 ENTER THE COMMAND LOOP.',/,T5,'THE 20 COMMANDS ARE
00104           2 LISTED IN THE INSTRUCTION NOTICE.',///)
00105   104     FORMAT (/,T10,'TYPE IN YOUR COMMAND CODE (1-20) : ',$)
00106   C
00107   200     FORMAT (I3)
00108   201     FORMAT (A10)
00109   202     FORMAT (E11.4)
00110   C
00111   C       END OF FORMAT BLOCK
00112   C
```

```
SELEC3  SELEC3.FOR      FORTRAN V.S(515) /KI    27-JUL-77       20:09   PAGE 1-2


00113   C
00114   C       FIRST COMMAND
00115   C
00116   1000    DO 1010 I=1,NCO
00117   1010    IND(I) = I                              ADD ALL COMPONENTS
00118           GO TO 3000                              UPDATE DISTANCES
00119   C
00120   C       SECOND COMMAND
00121   C
00122   1100    DO 1110 I=1,NCO
00123   1110    IND(I) = 0                              DELETE ALL COMPONENTS
00124           DO 1120 I=NDI,NDE
00125   1120    C(I) = 0.
00126           GO TO 20                                NEXT COMMAND
00127   C
00128   C       THIRD COMMAND
00129   C
00130   1200    TYPE 110
00131           ACCEPT 200,I                            GET COMPONENT SPECIFIC.
00132           IF (I .GT. NCO .OR. I .LE. 0) GO TO 1200
00133           IF (IND(I) .NE. 0) GO TO 1210
00134   C
00135   C       THE INPUT HAS NOW BEEN THOROUGHLY CHECKED
00136   C
00137           IND(I) = I                              ADD SPECIFIED COMP.
00138           GO TO 4000                              UPDATE DISTANCES
00139   C
00140   1210    TYPE 132                                COMP. ALREADY SELECTED
00141           GO TO 20
00142   C
00143   C       FOURTH COMMAND
00144   C
00145   1300    TYPE 111
00146           ACCEPT 200,I                            GET COMPONENT SPECIFIC.
00147           IF (I .GT. NCO .OR. I .LE. 0) GO TO 1300
00148           IF (IND(I) .EQ. 0) GO TO 1310
00149   C
00150   C       CHECK FINISHED
00151   C
00152           IND(I) = 0                              DELETE SPECIF. COMPON.
00153           GO TO 5000                              UPDATE DISTANCES
00154   C
00155   1310    TYPE 133                                COMP. ALREADY DELETED
00156           GO TO 20
00157   C
00158   C       FIFTH COMMAND
00159   C
00160   1400    DO 1410 I=NDE+ND+1,NCE
00161   1410    C(I) = 0.
00162           AT = 0.                                 TO AVERAGE CONTR.
00163           DO 1450 K=1,NCL-1
00164           DO 1450 L=K+1,NCL
00165           CALL UPDAT1(NCO,NDI,NDE,ND,NCL,K,L,AT)  UPDATE COMP. CONTRIB.
%FTNDIM LINE:00165 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00165 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00166   1450    CONTINUE                                NEXT CLASS
```

```
SELEC3  SELEC3.FOR      FORTRAN V.2(515) /KI    27-JUL-77       20:09   PAGE 1-3


00167   C
00168   C       THE CONTRIBUTION OF EACH UNSELECTED COMPONENT TO
00169   C       THE AVERAGE INTERCLASS DISTANCE HAS NOW BEEN COMPUTED
00170   C       THE LARGEST 'CONTRIBUTOR' WILL BE SELECTED
00171   C
00172           GO TO 1670                              USE PART OF OTHER ROUT.
00173   C
00174   C       SIXTH COMMAND
00175   C
00176   1500    DO 1510 I=NDE+ND+1,NCE
00177   1510    C(I) = 0.
00178           WT = 0.
00179           DO 1550 K=1,NCL-1
00180           DO 1550 L=K+1,NCL
00181           CALL UPDAT2(NCO,NDI,NDE,ND,NCL,K,L,WT)   UPDATE COMP. CONTRIB.
%FTNDIM LINE:00181 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00181 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00182   1550    CONTINUE
00183   C
00184   C       THE CONTRIBUTION OF EACH SELECTED COMPONENT TO
00185   C       THE AVERAGE INTERCLASS DISTANCE HAS NOW BEEN COMPUTED
00186   C       THE SMALLEST 'CONTRIBUTOR' WILL BE REJECTED
00187   C
00188           GO TO 1770                              USE PART OF OTHER ROUT.
00189   C
00190   C       SEVENTH COMMAND
00191   C
00192   1600    DO 1610 I=NDE+ND+1,NCE
00193   1610    C(I) = 0.
00194           WT = 0.
00195   1620    TYPE 112                                ASK FOR CLASS #
00196           ACCEPT 200,K                            GET IT
00197           IF (K .GT. NCL .OR. K .LE. 0) GO TO 1620
00198           DO 1650 L=1,NCL                         LOOP TO COMPUTE DIST.
00199           IF (L .EQ. K) GO TO 1650                SKIP CLASS ITSELF
00200           CALL UPDAT1(NCO,NDI,NDE,ND,NCL,K,L,WT)   UPDATE COMP. CONTRIB.
%FTNDIM LINE:00200 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00201   1650    CONTINUE                                NEXT CLASS
00202   C
00203   C       THIS LOOP HAS COMPUTED THE CONTRIBUTION OF EACH
00204   C       UNSELECTED COMPONENT TO THE TOTAL INTERCLASS DISTANCE
00205   C       NOW THE LARGEST 'CONTRIBUTOR' WILL BE SELECTED
00206   C
00207   1670    CALL MAX(I,NCO,NDE,ND)                  RETURNS THE INDEX IN I
00208           IF (I .EQ. 0) GO TO 1680                ERROR
00209           J = NDE+ND+I
00210           CONTR = C(J)/WT                         AVERAGE CONTRIBUTION
00211           IF (CONTR .LT. TA) GO TO 1690           IS IT BELOW THRESHOLD ?
00212           IND(I) = I                              NO, SO ADD COMPONENT
00213           TYPE 120,I,CONTR                        OUTPUT DATA
00214           GO TO 4000                              UPDATE DISTANCES
00215   C
00216   1680    TYPE 134                                COMMAND ERROR
00217   C
00218   C       VERY LIKELY, ALL COMPONENTS WERE ALREADY SELECTED
00219   C
```

```
SELEC3  SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-4


00220           GO TO 20
00221   C
00222   1690    TYPE 136                                BELOW THRESHOLD-TO-ADD
00223           GO TO 20
00224   C
00225   C       EIGHTH COMMAND
00226   C
00227   1700    DO 1710 I=NDE+ND+1,NCE
00228   1710    C(I) = 0.
00229           WT = 0.
00230   1720    TYPE 112                                ASK FOR CLASS #
00231           ACCEPT 200,K                            GET IT
00232           IF (K .GT. NCL .OR. K .LE. 0) GO TO 1720
00233           DO 1750 L=1,NCL
00234           IF (L .EQ. K) GO TO 1750                SKIP CLASS ITSELF
00235           CALL UPDAT2(NCO,NDI,NDE,ND,NCL,K,L,WT)   UPDATES COMP. CONTRIB.
%FTNDIM LINE:00235 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00236   1750    CONTINUE                                NEXT CLASS
00237   C
00238   C       THIS LOOP HAS COMPUTED THE CONTRIBUTION OF EACH SELECTED
00239   C       COMPONENT TO THE TOTAL INTERCLASS DISTANCE. THE SMALLEST
00240   C       'CONTRIBUTOR' WILL NOW BE DETERMINED AND REJECTED
00241   C
00242   1770    CALL MIN(I,NCO,NDE,ND)                  RETURNS INDEX IN I
00243           IF (I .EQ. 0) GO TO 1780                ERROR
00244           J = NDE+ND+I
00245           CONTR = C(J)/WT                         AVERAGE CONTRIBUTION
00246           IF (CONTR .GT. TR) GO TO 1790           IS IT ABOVE THRESHOLD ?
00247           IND(I) = 0                              NO, SO REJECT COMP.
00248           TYPE 121,I,CONTR                        DISPLAY INFORMATION
00249           GO TO 5000                              UPDATE DISTANCES
00250   C
00251   1780    TYPE 135                                COMMAND ERROR
00252   C
00253   C       VERY LIKELY, ALL COMPONENTS WERE ALREADY REJECTED
00254   C
00255           GO TO 20
00256   C
00257   1790    TYPE 137                                ABOVE THRESHOLD-TO-REJECT
00258           GO TO 20
00259   C
00260   C       NINTH COMMAND
00261   C
00262   1800    DO 1810 I=NDE+ND+1,NCE
00263   1810    C(I) = 0.
00264           WT = 0.
00265   1820    TYPE 113
00266           ACCEPT 203,K,L                          GET CLASS #S
00267           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00268           1 K .LE. 0 .OR. L .LE. 0) GO TO 1820
00269           IF (K .EQ. L) GO TO 2060
00270           CALL UPDAT1(NCO,NDI,NDE,ND,NCL,K,L,WT)   UPDATE COMP. CONTRIB.
00271           GO TO 1670                              SELECT COMPONENT
00272   C
00273   C       TENTH COMMAND
00274   C
```

```
SELEC3  SELEC3.FOR      FORTRAN V.S(515) /KI    27-JUL-77       20:09   PAGE 1-5


00275   1900    DO 1910 I=NDE+ND+1,NCE
00276   1910    C(I) = 0.
00277           WT = 0.
00278   1920    TYPE 113
00279           ACCEPT 203,K,L                          GET CLASS #S
00280           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00281           2 K .LE. 0 .OR. L .LE. 0) GO TO 1920
00282           IF (K .EQ. L) GO TO 2060
00283           CALL UPDAT2(NCO,NDI,NDE,ND,NCL,K,L,WT)   UPDATE COMP. CONTRIB.
00284           GO TO 1770                              DESELECT COMPONENT
00285   C
00286   C       ELEVENTH COMMAND
00287   C
00288   2000    TYPE 113
00289           ACCEPT 203,K,L                          GET CLASS #S
00290           IF (K .GT. NCL .OR. L .GT. NCL .OR.
00291           1 K .LE. 0 .OR. L .LE. 0) GO TO 2000
00292           IF (K .EQ. L) GO TO 2060
00293           NORD1 = (NCL-1)*(K-1)+L-1+NDI-1
00294           NORD2 = (NCL-1)*(L-1)+K-1+NDI-1  INDEX #S IN DIST
00295           IF (K .GT. L) NORD1=NORD1+1
00296           IF (K .LT. L) NORD2=NORD2+1
00297           HMS = 0.
00298           IF (C(NORD1) .EQ. 0. .AND. C(NORD2) .EQ. 0.) GO TO 2010
00299           HMS =  4.*C(NORD1)*C(NORD2)/((SQRT(C(NORD1))+SQRT(C(NORD2)))**2)
00300   C
00301   C       NORD1 & 2 ARE THE DISPLACEMENTS IN THE ARRAY C (DIST.)
00302   C       HMS STANDS FOR HARMONIC MEAN SQUARE
00303   C
00304   2010    TYPE 122,K,L,K,C(NORD1),L,C(NORD2),HMS
00305           GO TO 20                                NEXT COMMAND
00306   C
00307   2060    TYPE 123                                ERROR MESSAGE
00308           GO TO 20
00309   C
00310   C       TWELFTH COMMAND
00311   C
00312   2100    TYPE 124
00313           CALL PACK(NCO,NCE,NDE,ND,J)
00314           TYPE 125,(C(I),I=NDE+ND+1,NDE+ND+J)      TYPE SELECTED COMP. #S
00315           TYPE 130
00316           GO TO 20                                NEXT COMMAND
00317   C
00318   C       THIRTEENTH COMMAND
00319   C
00320   2200    CALL PARAM(NCO,NCL,ND,NDI,NDE,NSB,VMIN,AVER,SD,I,J)
00321           TYPE 126,NSB,AVER,SD,VMIN,I,J,IA,TR      SHORT STATUS
00322           GO TO 20                                NEXT COMMAND
00323   C
00324   C       FOURTEENTH COMMAND
00325   C
00326   2300    CALL PARAM(NCO,NCL,ND,NDI,NDE,NSB,VMIN,AVER,SD,I,J)
00327           PRINT 126,NSB,AVER,SD,VMIN,I,J,IA,TR     SHORT STATUS
00328           PRINT 127
00329           CALL PACK(NCO,NCE,NDE,ND,J)
00330           PRINT 125,(C(I),I=NDE+ND+1,NDE+ND+J)     LIST OF SEL. COMP.
```

```
SELEC3  SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-6


00331           PRINT 130
00332           GO TO 20                                NEXT COMMAND
00333   C
00334   C       FIFTEENTH COMMAND
00335   C
00336   2400    CALL PARAM(NCO,NCL,ND,NDI,NDE,NSB,VMIN,AVER,SD,I,J)
00337           PRINT 120,NSB,AVER,SD,VMIN,I,J,TA,TR      SHORT STATUS
00338           PRINT 127
00339           CALL PACK(NCO,NCE,NDE,ND,J)
00340           PRINT 125,(C(I),I=NDE+ND+1,NDE+ND+J)      LIST OF SEL. COMP.
00341           PRINT 130
00342           PRINT 128
00343           K = 0
00344   C
00345   C       THE PROGRAM WILL NOW PRINT ALL INTERCLASS DISTANCES,
00346   C       ORDERED BY CLASS
00347   C
00348           DO 2410 I=1,NCL
00349           DO 2410 J=1,NCL
00350           IF (I .EQ. J) GO TO 2410                NOT A POSSIBLE DISTANCE
00351           NORD1 = (NCL-1)*(I-1)+J-1+NDI-1
00352           IF (I .GT. J) NORD1=NORD1+1
00353           NORD2 = NORD1+ND
00354           IF (K .NE. 0) GO TO 2405                1ST OR 2ND PRINT COLUMN
00355           K = 1
00356           PRINT 131                               1ST COLUMN
00357           PRINT 129,I,J,C(NORD1),C(NORD2)
00358           GO TO 2410
00359   C
00360   2405    K = 0
00361           PRINT 129,I,J,C(NORD1),C(NORD2)             2ND COLUMN
00362   C
00363   2410    CONTINUE
00364           PRINT 130
00365           GO TO 20                                NEXT COMMAND
00366   C
00367   C       SIXTEENTH COMMAND
00368   C
00369   2500    TYPE 114                                ASK FOR THRESHOLD-TO-ADD
00370           ACCEPT 204,TA
00371           GO TO 20
00372   C
00373   C       SEVENTEENTH COMMAND
00374   C
00375   2600    TYPE 115                                ASK FOR THRESHOLD-TO-REJECT
00376           ACCEPT 204,TR
00377           GO TO 20
00378   C
00379   C       EIGHTEENTH COMMAND
00380   C
00381   2700    TYPE 116                                ASK FOR CLASS #S,#EIGHT
00382           ACCEPT 205,I,J,*
00383           IF (I .LE. 0 .OR. I .GT. NCL .OR.
00384          1 J .LE. 0 .OR. J .GT. NCL) GO TO 2700
00385           IF (I .EQ. J) GO TO 2050
00386           NORD1 = (NCL-1)*(I-1)+J-1+NDI-1+ND
```

```
SELEC3  SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1-7

00387           NORD2 = (NCL-1)*(J-1)+I-1+NDI-1+ND
00388           IF (I .GT. J) NORD1=NORD1+1
00389           IF (I .LT. J) NORD2=NORD2+1
00390           C(NORD1) = R
00391           C(NORD2) = R
00392           GO TO 20
00393   C
00394   C       NINETEENTH COMMAND
00395   C
00396   2800    TYPE 113                                ASK FOR CLASS #S
00397           ACCEPT 203,I,J
00398           IF (I .LE. 0 .OR. I .GT. NCL .OR.
00399           1 J .LE. 0 .OR. J .GT. NCL) GO TO 2800
00400           IF (I .EQ. J) GO TO 2050
00401           NORD1 = (NCL-1)*(I-1)+J-1+NDI-1+ND
00402           IF (I .GT. J) NORD1=NORD1+1
00403           TYPE 138,C(NORD1)                       TYPE WEIGHT FOR THESE CLASSES
00404           GO TO 20
00405   C
00406   C       TWENTIETH COMMAND
00407   C
00408   2900    STOP                                    END OF PROGRAM
00409   C
00410   C       DISTANCE UPDATING (ALL COMPONENTS)
00411   C       DONE BY EVALUATING ALL QUADRATIC FORMS COMPLETELY
00412   C
00413   3000    DO 3100 I=1,NCL-1
00414           DO 3100 J=I+1,NCL
00415           NORD1 = (NCL-1)*(I-1)+J-1+NDI-1
00416           NORD2 = (NCL-1)*(J-1)+I-1+NDI-1
00417           IF (I .GT. J) NORD1=NORD1+1
00418           IF (I .LT. J) NORD2=NORD2+1
00419           C(NORD1) = 0.
00420           C(NORD2) = 0.
00421   C
00422   C       DIST IS INITIALIZED AT ZERO
00423   C
00424           DO 3100 K=1,NCO
00425           PROD1 = 0.
00426           PROD2 = 0.
00427           DO 3050 L=1,NCO
00428           I1 = NC*(I-1)+K*NCC+L
00429           I2 = NC*(J-1)+K*NCC+L
00430           L1 = NC*(I-1)+L
00431           L2 = NC*(J-1)+L
00432           PROD1 = PROD1+C(I1)*(C(L1)-C(L2))
00433           PROD2 = PROD2+C(I2)*(C(L1)-C(L2))       ACC. PART. PROD.
00434   3050    CONTINUE
00435           L1 = NC*(I-1)+K
00436           L2 = NC*(J-1)+K
00437           C(NORD1) = C(NORD1)+PROD1*(C(L1)-C(L2))
00438           C(NORD2) = C(NORD2)+PROD2*(C(L1)-C(L2))
00439   C
00440   C       ACCUMULATES PARTIAL SUMS
00441   C
00442   3100    CONTINUE
```

```
SELEC3  SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1-8


00443            GO TO 20                                       NEXT COMMAND
00444    C
00445    C       DISTANCE UPDATING (I IS INDEX OF ADDED COMPONENT)
00446    C       DONE BY ADDING THE NEW COMPONENT'S CONTRIBUTIONS TO THE
00447    C       PREVIOUS DISTANCES (AS FOUND IN THE ARRAY DIST)
00448    C
00449    4000    DO 4100 J=1,NCL-1
00450            DO 4100 K=J+1,NCL
00451            CALL TEMP(NCO,NCL,NDI,I,J,K,NORD1,NORD2,PROD1,PROD2)
%FTNDIM LINE:00451 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00451 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00452            I1 = NC*(J-1)+I
00453            I2 = NC*(K-1)+I
00454            L1 = NC*(J-1)+I*(NCO+1)
00455            L2 = NC*(K-1)+I*(NCO+1)
00456            C(NORD1) = C(NORD1)+(C(I1)-C(I2))*(PROD1+C(L1)*
00457            1(C(I1)-C(I2)))
00458            C(NORD2) = C(NORD2)+(C(I1)-C(I2))*(PROD2+C(L2)*
00459            1(C(I1)-C(I2)))
00460    4100    CONTINUE
00461            GO TO 20                                       NEXT COMMAND
00462    C
00463    C       DISTANCE UPDATING (I IS INDEX OF DELETED COMPONENT)
00464    C       DONE BY SUBTRACTING THE REJECTED COMPONENT'S CONTRIBUTION
00465    C       FROM THE PREVIOUS DISTANCES ( AS FOUND IN ARRAY DIST)
00466    C
00467    5000    DO 5100 J=1,NCL-1
00468            DO 5100 K=J+1,NCL
00469            CALL TEMP(NCO,NCL,NDI,I,J,K,NORD1,NORD2,PROD1,PROD2)
%FTNDIM LINE:00469 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
%FTNDIM LINE:00469 POSSIBLE DO INDEX MODIFICATION INSIDE LOOP
00470            I1 = NC*(J-1)+I
00471            I2 = NC*(K-1)+I
00472            L1 = NC*(J-1)+I*(NCO+1)
00473            L2 = NC*(K-1)+I*(NCO+1)
00474            C(NORD1) = C(NORD1)-(C(I1)-C(I2))*(PROD1+C(L1)*
00475            1(C(I1)-C(I2)))
00476            C(NORD2) = C(NORD2)-(C(I1)-C(I2))*(PROD2+C(L2)*
00477            1(C(I1)-C(I2)))
00478    5100    CONTINUE
00479            GO TO 20                                       NEXT COMMAND
00480    C
00481    C
00482    C       FORMAT BLOCK
00483    C
00484    110     FORMAT (T10,'TYPE IN THE # OF THE COMP. TO BE ADDED : ',$)
00485    111     FORMAT (T10,'TYPE IN THE # OF THE COMP. TO BE DELETED : ',$)
00486    112     FORMAT (T10,'TYPE IN CLASS # : ',$)
00487    113     FORMAT (T10,'TYPE IN BOTH CLASS NUMBERS, SEP. BY A COMMA : ',$)
00488    114     FORMAT (T10,'TYPE IN THRESHOLD-TO-ADD : ',$)
00489    115     FORMAT (T10,'TYPE IN THRESHOLD-TO-REJECT : ',$)
00490    116     FORMAT (T10,'TYPE IN BOTH CLASS #S & WEIGHT, SEP.
00491            1 BY A COMMA : ',$)
00492    C
00493    120     FORMAT (T10,'THE SELECTED COMP. IS #',I3,', WITH A
00494            1 CONTR. OF ',E11.4,/)
```

```
SELEC3  SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1-9


00495    121      FORMAT (T10,'THE REJECTED COMP. IS #',I3,', WHICH HAD A
00496              1 CONTR. OF ',E11.4,/)
00497    122      FORMAT (T10,'THE DISTANCES BETWEEN CLASS',I3,' AND CLASS'
00498              1,I3,' ARE :',/,T10,'--REL. TO CLASS',I3,' : ',E11.4,/,
00499              2T10,'--REL. TO CLASS',I3,' : ',E11.4,/,T10,
00500              3'--HARMONIC MEAN SQUARE : ',E11.4,/)
00501    123      FORMAT (T10,'CLASS #S MUST BE DIFFERENT  ',/)
00502    124      FORMAT (T10,'THE FOLLOWING COMPONENTS ARE SELECTED :')
00503    125      FORMAT (/,10(3X,F4.0))
00504    126      FORMAT (T10,'THE PRESENT STATUS IS :',/,T10,
00505              1'--NUMBER OF SELECTED COMPONENTS :',I4,/,T10,
00506              2'--AVERAGE INTERCLASS DISTANCE : ',E11.4,/,T10,
00507              3'  WITH A  STANDARD DEV. OF    : ',E11.4,/,T10,
00508              4'--THE MINIMUM RMS DISTANCE IS : ',E11.4,/,T10,
00509              5'  BETWEEN CLASSES',I3,' AND',I3,/,T10,
00510              6'--THE THRESHOLD-TO-ADD IS      : ',E11.4,/,T10,
00511              7'  THE THRESHOLD-TO-REJECT IS  : ',E11.4,/)
00512    127      FORMAT (T10,'THE SELECTED COMPONENTS ARE :')
00513    128      FORMAT (/,T10,'THE INTERCLASS DISTANCES ARE :',//,T10,
00514              12('FROM:  TO:  DISTANCE:   WEIGHT:     '))
00515    129      FORMAT (4X,I3,3X,I3,2X,E11.4,2X,F6.3,3X,$)
00516    130      FORMAT (//)
00517    131      FORMAT (/,T7,$)
00518    132      FORMAT (/,T10,'THIS COMPONENT IS ALREADY SELECTED  ',/)
00519    133      FORMAT (/,T10,'THIS COMPONENT IS ALREADY REJECTED  ',/)
00520    134      FORMAT (/,T10,'ERROR : ALL COMPONENTS ALREADY SELECTED
00521              1  ',/)
00522    135      FORMAT (/,T10,'ERROR : ALL COMPONENTS ALREADY REJECTED
00523              1  ',/)
00524    136      FORMAT (/,T10,'CONTRIB. BELOW THRESHOLD-TO-ADD  ',/)
00525    137      FORMAT (/,T10,'CONTRIB. ABOVE THRESHOLD-TO-REJECT  ',/)
00526    138      FORMAT (/,T10,'THE WEIGHT IS : ',F6.3,/)
00527    C
00528    203      FORMAT (2I)
00529    204      FORMAT (F10.5)
00530    205      FORMAT (2I,F5.3)
00531    C
00532    C        END OF FORMAT BLOCK
00533    C        AND END OF PROGRAM
00534             END


COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210


SUBPROGRAMS CALLED

MIN     SQRT.   MAX     UPDAT1  UPDAT2  TEMP
PACK    PARAM
```

```
SELEC3  SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1-10


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

 .I0013 1        .S0020 2        .I0012 3        .I0011 4        *NCO    5
*NORD1  6        .I0010 7       *SD     10       *W      11      *NDE    12
 .S0037 13       *NS    14       .S0036 15        .S0035 16       .S0034 17
 .S0033 20        .S0032 21      *K      22       .S0031 23      *NDI    24
 .S0030 25       *NCL   26       *TH     27      *TA     30      *NCE    31
 .S0040 32       *HMS   33       *MS     34      *NCM    35      *J      36
 .S0007 37        .S0006 40       .S0005 41        D      42      *NSB    112
 .S0004 113       .S0003 114      .S0002 115       .S0001 116       .S0000 117
*L2     120      *I2    121      *AVER   122       .S0017 123       .S0016 124
 .S0015 125       .I0007 126      .S0014 127      *NU     130       .I0006 131
 .S0013 132       .I0005 133      .S0012 134      *PROD2  135       .I0004 136
 .S0011 137      *L1    140       .I0003 141       .S0010 142      *L      143
 .I0002 144      *I1    145       .I0001 146      *NORD2  147      *NT     150
 .I0000 151      *I     152      *VMIN   153       .S0027 154       .S0026 155
 .S0025 156      *NC    157       .I0017 160       .S0024 161       .I0016 162
 .S0023 163       .I0015 164      .S0022 165      *PROD1  166       .I0014 167
 .S0021 170      *CONTR 171

TEMPORARIES

 .00000 1050     .00001 1051

%FTNWRN   SELEC3        NO FATAL ERRORS AND 10 WARNINGS
```

```
MAIN.   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MAX(I,NCO,NDE,ND)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010           I = 0
00011           VMAX = -1.E25
00012           DO 10   J=1,NCO
00013           IF (IND(J) .NE. 0) GO TO 10                     ONLY UNSEL. COMP.
00014           K = NDE+ND+J
00015           IF (VMAX .GE. C(K)) GO TO 10                    TEST FOR MAX.
00016           I = J
00017           VMAX = C(K)                                     FOUND A LARGER
00018   10      CONTINUE
00019           RETURN
00020           END

COMMON BLOCKS

/.COMM./(+1103+3)
C       +0      IND     +110210

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *NDE    2       *VMAX   3       *K      4       *J      5
 .S0000 6       *ND     7       *I      10

TEMPORARIES


MAX       NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.3(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE MIN(I,NCO,NDE,ND)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010           I = 0
00011           VMIN = 1.E+26
00012           DO 10   J=1,NCO
00013           IF (IND(J) .EQ. 0) GO TO 10             ONLY SEL. COMP.
00014           K = NDE+ND+J
00015           IF (VMIN .LE. C(K)) GO TO 10            TEST FOR MIN.
00016           I = J
00017           VMIN = C(K)                             FOUND A LARGER
00018   10      CONTINUE
00019           RETURN
00020           END

COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +11021U

SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *NDE    2       *K      3       *J      4       .50000 5
*ND     6       *I      7       *VMIN   10

TEMPORARIES


MIN       NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE PARAM(NCO,NCL,NO,NDI,NDE,NSB,VMIN,AVER,SD,L,M)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010   C
00011   C       THIS ROUTINE WILL COUNT THE NUMBER OF SEL. COMPON.
00012   C       AND COMPUTE THE AVERAGE INTERCLASS DISTANCE, ITS
00013   C       STANDARD DEVIATION AND ITS MINIMUM PART.
00014   C
00015           NSB = 0
00016           DO 10 K=1,NCO
00017           IF (IND(K) .NE. 0) NSB=NSB+1
00018   10      CONTINUE
00019           AVER = 0.
00020           L = 1
00021           M = 2
00022           VMIN = 1.E26
00023           DO 20 K=NDI,NDE
00024           AVER = AVER+C(K)/NO
00025           J = K+1-NDI
00026           I = (J-1)/(NCL-1)+1
00027           J = J-(NCL-1)*(I-1)
00028           IF (J .GE. I) J=J+1
00029           NORD2 = (NCL-1)*(J-1)+I-1+NDI-1
00030           IF (J .GT. I) NORD2=NORD2+1
00031           HMS = 0.
00032           IF (C(K) .EQ. 0. .AND. C(NORD2) .EQ. 0.) GO TO 15
00033           HMS = 4.*C(K)*C(NORD2)/((SQRT(C(K))+SQRT(C(NORD2)))**2)
00034   15      IF (VMIN .LE. HMS) GO TO 20
00035           VMIN = HMS
00036           L = I
00037           M = J
00038   20      CONTINUE
00039           SD = 0.
00040           DO 30 I=NDI,NDE
00041   30      SD = SD+((C(I)-AVER)**2)/NO
00042           SD = SQRT(SD)
00043           RETURN
00044           END

COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210

SUBPROGRAMS CALLED

SQRT.
```

```
PARAM   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-1


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1       *SD      2       *NOE     3       *K       4       *NOI     5
*NCL     6       *HMS     7       *M       10      *J       11      *NSB     12
 .S0002 13        .S0001 14        .S0000 15       *AVER    16      *ND      17
*L       20      *NORD2   21      *I       22      *VMIN    23

TEMPORARIES

 .A0016 24        .Q0000 25        .Q0001 26

PARAM      NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1


00001    C
00002    C
00003    C
00004            SUBROUTINE UPDAT2(NCO,NDI,NDE,ND,NCL,K,L,WT)
00005            PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006            COMMON C(1:NDIM)
00007            COMMON IND(1:NCOMP)
00008            REAL C
00009            INTEGER IND
00010    C
00011    C       THIS ROUTINE WILL COMPUTE THE CONTRIBUTION OF SELECTED
00012    C       COMPONENTS, ONE STEP AT A TIME (ADDING ONE CONTRIBUTION
00013    C       EVERY TIME IT'S CALLED)
00014    C
00015            NC = NCO*(NCO+1)
00016            NORD = (NCL-1)*(K-1)+L-1+NDI-1+ND
00017            IF (K .GT. L) NORD=NORD+1
00018            WT = WT+2.*C(NORD)
00019            DO 100 I=1,NCO
00020            IF (IND(I) .EQ. 0) GO TO 100      ONLY SEL. COMP.
00021            PROD = 0.
00022            DO 50 J=1,NCO
00023            IF (J .EQ. I) GO TO 50
00024            IF (IND(J) .EQ. 0) GO TO 50       USE ONLY SEL. COMP.
00025            J1 = NC*(K-1)+J
00026            J2 = NC*(L-1)+J
00027            I1 = NC*(K-1)+I*NCO+J
00028            I2 = NC*(L-1)+I*NCO+J
00029            PROD = PROD+2*(C(J1)-C(J2))*(C(I1)+C(I2))
00030    50      CONTINUE
00031            J1 = NC*(K-1)+I
00032            J2 = NC*(L-1)+I
00033            I1 = NC*(K-1)+I*(NCO+1)
00034            I2 = NC*(L-1)+I*(NCO+1)
00035            M = NDE+ND+I
00036            C(M) = C(M)+C(NORD)*(C(J1)-C(J2))*(PROD+(C(I1)+C(I2))*
00037            1(C(J1)-C(J2)))
00038    100     CONTINUE
00039            RETURN
00040            END


COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210


SUBPROGRAMS CALLED
```

```
UPDAT2  SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-1


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO     1      *NDE     2      *K       3      *NDI     4      *NCL     5
*PROD    6      *NORD    7      *J2      10     *M       11     *J1      12
*J       13      .S0001  14      .S0000  15     *I4      16     *ND      17
*L       20     *I1      21     *I       22     *WT      23     *NC      24

TEMPORARIES


UPDAT2    NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.S(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE UPDAT1(NCO,NOI,NDE,ND,NCL,K,L,WT)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010   C
00011   C       THIS ROUTINE WILL COMPUTE THE CONTRIBUTION OF UNSELECTED
00012   C       COMPONENTS, ONE STEP AT A TIME (ADDING ONE CONTRIBUTION
00013   C       EVERY TIME IT'S CALLED)
00014   C
00015           NC = NCO*(NCO+1)
00016           NORD = (NCL-1)*(K-1)+L-1+NOI-1+ND
00017           IF (K .GT. L) NORD=NORD+1
00018           WT = WT+2.*C(NORD)
00019           DO 100 I=1,NCO
00020           IF (IND(I) .NE. 0) GO TO 100      ONLY UNSEL. COMP.
00021           PROD = 0.
00022           DO 50 J=1,NCO
00023           IF (IND(J) .EQ. 0) GO TO 50       USE ONLY SEL. COMP.
00024           J1 = NC*(K-1)+J
00025           J2 = NC*(L-1)+J
00026           I1 = NC*(K-1)+I*NCO+J
00027           I2 = NC*(L-1)+I*NCO+J
00028           PROD = PROD+2*(C(J1)-C(J2))*(C(I1)+C(I2))
00029   50      CONTINUE
00030           J1 = NC*(K-1)+I
00031           J2 = NC*(L-1)+I
00032           I1 = NC*(K-1)+I*(NCO+1)
00033           I2 = NC*(L-1)+I*(NCO+1)
00034           M = NDE+ND+I
00035           C(M) = C(M)+C(NORD)*(C(J1)-C(J2))*(PROD+(C(I1)+C(I2))*
00036           1(C(J1)-C(J2)))
00037   100     CONTINUE
00038           RETURN
00039           END


COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *NDE    2       *K      3       *NOI    4       *NCL    5
*PROD   6       *NORD   7       *J2     10      *M      11      *J1     12
```

```
UPDAT1  SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-1

*J      13      .S0001 14      .S0000 15      *I2     16      *ND     17
*L      20      *I1    21      *I      22      *WT     23      *NC     24

TEMPORARIES


UPDAT1    NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE PACK(NCO,NCE,NDE,ND,J)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010   C
00011   C       THIS ROUTINE WILL 'COMPACTIFY' THE ARRAY CONT, I.E.
00012   C       WILL JUSTIFY IT TO THE BEGINNING
00013   C
00014           DO 5 I=NDE+ND+1,NCE
00015   5       C(I) = 0.
00016           J = 0
00017           DO 10 I=1,NCO
00018           IF (IND(I) .EQ. 0) GO TO 10      SKIP UNSEL. COMP.
00019           J = J+1
00020           K = NDE+ND+J
00021           C(K) = I
00022   10      CONTINUE
00023           RETURN
00024           END


COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *NDE    2       *K      3       *NCE    4       *J      5
 .S0001 6        .S0000 7       *ND     10       .I0000 11      *I      12


TEMPORARIES


PACK      NO ERRORS DETECTED
```

```
MAIN.   SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1


00001   C
00002   C
00003   C
00004           SUBROUTINE TEMP(NCO,NCL,NDI,I,J,K,NORD1,NORD2,PROD1,PROD2)
00005           PARAMETER NCLASS=20,NCOMP=91,NDIM=37000
00006           COMMON C(1:NDIM)
00007           COMMON IND(1:NCOMP)
00008           REAL C
00009           INTEGER IND
00010   C
00011   C       THIS ROUTINE HELPS WITH THE COMPUTATION OF COMPONENTS
00012   C       CONTRIBUTIONS
00013   C
00014           NC = NCO*(NCO+1)
00015           NORD1 = (NCL-1)*(J-1)+K-1+NDI-1
00016           NORD2 = (NCL-1)*(K-1)+J-1+NDI-1
00017           IF (J .GT. K) NORD1=NORD1+1
00018           IF (J .LT. K) NORD2=NORD2+1
00019           PROD1 = 0.
00020           PROD2 = 0.
00021           DO 10  L=1,NCO
00022           IF (IND(L) .EQ. 0) GO TO 10       ONLY SEL. COMP.
00023           IF (L .EQ. I) GO TO 10            SKIP THIS INDEX
00024           I1 = NC*(J-1)+L
00025           I2 = NC*(K-1)+L
00026           J1 = NC*(J-1)+I*NCO+L
00027           J2 = NC*(K-1)+I*NCO+L
00028           PROD1 = PROD1+2*(C(I1)-C(I2))*C(J1)
00029           PROD2 = PROD2+2*(C(I1)-C(I2))*C(J2)
00030   10      CONTINUE
00031           RETURN
00032           END


COMMON BLOCKS

/.COMM./(+110343)
C       +0      IND     +110210


SUBPROGRAMS CALLED


SCALARS AND ARRAYS   "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED

*NCO    1       *NORD1  2       *K      3       *NDI    4       *NCL    5
*J2     6       *J1     7       *J      10      *S0000  11      *I2     12
*PROD2  13      *L      14      *I1     15      *NORD2  16      *I      17
*NC     20      *PROD1  21
```

```
TEMP    SELEC3.FOR      FORTRAN V.5(515) /KI    27-JUL-77       20:09   PAGE 1-1

TEMPORARIES

TEMP       NO ERRORS DETECTED
```

SAMPLE PROGRAM PRINTOUT

THE PROBLEM USED IS THE RADAR RETURN CLASSIFICATION PROBLEM, I.E., 8 CLASSES, 33 INITIAL MEASUREMENTS. THE CLASSES ARE NUMBERED FROM 1 TO 8 AS FOLLOWS :

1 -- CSA 'GALAXY'
2 -- AN-22
3 -- B737
4 -- TU-22
5 -- FB111A
6 -- MIG-25
7 -- A4M 'SKYHAWK'
8 -- MIG-21

THE FIRST TWO PAGES ARE A PRINTOUT OF THE PROGRAM SELEC2.FOR USED TO OBTAIN A FEATURE ORDERING. WEIGHTS LARGER THAN ONE WERE ASSIGNED TO THE DISTANCES BETWEEN PAIRED AIRCRAFT, AS SHOWN IN THE STATUS PRINTOUT.

THE NEXT 16 PAGES ARE TWO PRINTOUTS OF THE PROGRAM CLASS1.FOR I.E., CLASSIFICATION RESULTS. THE FIRST 8 PAGES SHOW THE RESULTS OBTAINED WITH THE FIRST 24 COMPONENTS AS LISTED ABOVE, WHILE THE NEXT 8 PAGES SHOW THE RESULTS OBTAINED WITH ALL COMPONENTS.
THE CLASSIFICATION RESULTS ARE SHOWN CLASS PER CLASS, THE COLUMNS LISTING THE DISTANCES FROM ONE SAMPLE IN THE CLASS TO ALL CLASSES; THE NUMBER IN THE FIRST COLUMN IS THE CLASSIFICATION, I.E., THE NUMBER OF THE CLASS TO WHICH THE PATTERN OF THIS LINE IS ASSIGNED.

APPENDIX D

SAMPLE RUNS

THE SELECTED COMP. IS # 25, WITH A CONTR. OF 0.1125E+07
THE SELECTED COMP. IS # 12, WITH A CONTR. OF 0.1625E+08
THE SELECTED COMP. IS # 3, WITH A CONTR. OF 0.6236E+08
THE SELECTED COMP. IS # 19, WITH A CONTR. OF 0.2170E+08
THE SELECTED COMP. IS # 8, WITH A CONTR. OF 0.4155E+08
THE SELECTED COMP. IS # 1, WITH A CONTR. OF 0.8343E+08
THE SELECTED COMP. IS # 23, WITH A CONTR. OF 0.1864E+08
THE SELECTED COMP. IS # 10, WITH A CONTR. OF 0.7905E+08
THE SELECTED COMP. IS # 21, WITH A CONTR. OF 0.7917E+08
THE SELECTED COMP. IS # 33, WITH A CONTR. OF 0.1835E+08
THE SELECTED COMP. IS # 26, WITH A CONTR. OF 0.1061E+08
THE SELECTED COMP. IS # 4, WITH A CONTR. OF 0.3463E+08
THE SELECTED COMP. IS # 15, WITH A CONTR. OF 0.1318E+08
THE SELECTED COMP. IS # 30, WITH A CONTR. OF 0.6040E+08
THE SELECTED COMP. IS # 13, WITH A CONTR. OF 0.4208E+08
THE SELECTED COMP. IS # 28, WITH A CONTR. OF 0.4050E+08
THE SELECTED COMP. IS # 6, WITH A CONTR. OF 0.2288E+08
THE SELECTED COMP. IS # 31, WITH A CONTR. OF 0.1505E+08
THE SELECTED COMP. IS # 16, WITH A CONTR. OF 0.2080E+09
THE SELECTED COMP. IS # 22, WITH A CONTR. OF 0.1846E+10
THE SELECTED COMP. IS # 11, WITH A CONTR. OF 0.4105E+09
THE SELECTED COMP. IS # 18, WITH A CONTR. OF 0.2616E+09
THE SELECTED COMP. IS # 5, WITH A CONTR. OF 0.3558E+09
THE SELECTED COMP. IS # 7, WITH A CONTR. OF 0.9961E+08
THE SELECTED COMP. IS # 29, WITH A CONTR. OF 0.1333E+09
THE SELECTED COMP. IS # 24, WITH A CONTR. OF 0.2486E+09
THE SELECTED COMP. IS # 14, WITH A CONTR. OF 0.5218E+09
THE SELECTED COMP. IS # 27, WITH A CONTR. OF 0.4465E+09
THE SELECTED COMP. IS # 17, WITH A CONTR. OF 0.9216E+09
THE SELECTED COMP. IS # 32, WITH A CONTR. OF 0.6951E+09
THE SELECTED COMP. IS # 20, WITH A CONTR. OF 0.5235E+09
THE SELECTED COMP. IS # 9, WITH A CONTR. OF 0.3439E+10
THE SELECTED COMP. IS # 2, WITH A CONTR. OF 0.3333E+09

THE PRESENT STATUS IS :
--NUMBER OF SELECTED COMPONENTS : 33
--AVERAGE INTERCLASS DISTANCE : 0.9995E+11
WITH A STANDARD DEV. OF : 0.6156E+12
--THE MINIMUM RMS DISTANCE IS : 0.4254E+02
BETWEEN CLASSES 3 AND 5
--THE THRESHOLD-TO-ADD IS : 0.0000E+00
THE THRESHOLD-TO-REJECT IS : 0.0000E+00

THE SELECTED COMPONENTS ARE :

1.	2.	3.	4.	5.	6.	7.	8.	9.	10.
11.	12.	13.	14.	15.	16.	17.	18.	19.	20.
21.	22.	23.	24.	25.	26.	27.	28.	29.	30.
31.	32.	33.							

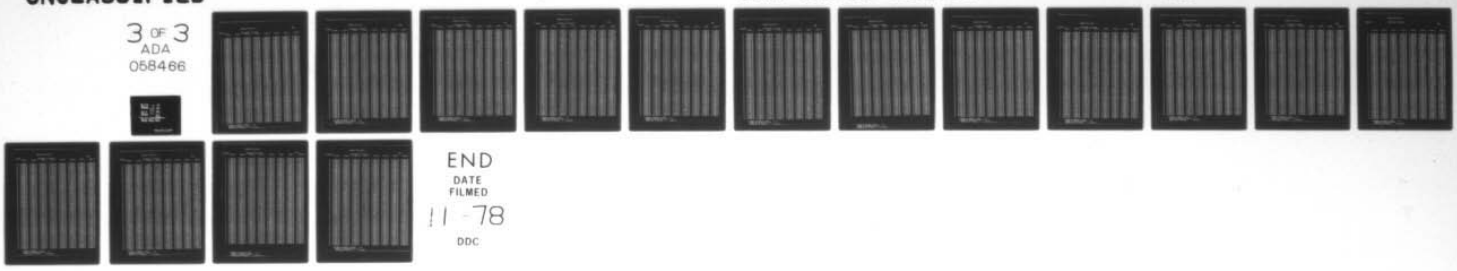
THE INTERCLASS DISTANCES ARE :

FROM:	TO:	DISTANCE:	WEIGHT:	FROM:	TO:	DISTANCE:	WEIGHT:
1	2	0.1165E+03	10.000	1	3	0.1358E+03	1.000
1	4	0.9441E+02	1.000	1	5	0.1481E+03	1.000
1	6	0.1618E+03	1.000	1	7	0.1936E+03	1.000
1	8	0.1887E+03	1.000	2	1	0.5155E+05	10.000
2	3	0.2101E+03	1.000	2	4	0.4084E+03	1.000
2	5	0.2726E+03	1.000	2	6	0.3152E+03	1.000
2	7	0.3894E+03	1.000	2	8	0.3976E+03	1.000
3	1	0.1380E+09	1.000	3	2	0.1395E+07	1.000
3	4	0.2789E+07	20.000	3	5	0.1154E+02	1.000
3	6	0.9180E+02	1.000	3	7	0.3619E+03	1.000
3	8	0.4524E+03	1.000	4	1	0.3395E+05	1.000
4	2	0.1160E+04	1.000	4	3	0.1114E+03	20.000
4	5	0.1664E+02	1.000	4	6	0.1586E+03	1.000
4	7	0.5982E+03	1.000	4	8	0.7189E+03	1.000
5	1	0.2312E+10	1.000	5	2	0.1454E+09	1.000
5	3	0.6629E+04	1.000	5	4	0.3006E+08	1.000
5	6	0.2421E+02	99.000	5	7	0.1302E+03	1.000
5	8	0.1320E+03	1.000	6	1	0.3137E+10	1.000
6	2	0.6990E+08	1.000	6	3	0.1391E+05	1.000
6	4	0.1699E+09	1.000	6	5	0.9196E+03	99.000
6	7	0.2637E+02	1.000	6	8	0.1632E+02	1.000
7	1	0.5857E+12	1.000	7	2	0.7286E+10	1.000
7	3	0.4721E+07	1.000	7	4	0.1854E+11	1.000
7	5	0.3255E+06	1.000	7	6	0.3506E+06	1.000
7	8	0.2788E+02	99.000	8	1	0.4624E+13	1.000
8	2	0.1271E+12	1.000	8	3	0.8545E+07	1.000
8	4	0.2283E+12	1.000	8	5	0.6767E+07	1.000
8	6	0.2872E+06	1.000	8	7	0.1862E+03	99.000

AD-A058 466 TENNESSEE UNIV KNOXVILLE DEPT OF ELECTRICAL ENGINEERING F/G 9/2
A RECURSIVE PROCEDURE FOR UPDATING QUADRATIC FORMS AND ITS APPL--ETC(U)
JAN 78 R C GONZALEZ, B M MORET N00014-75-C-0545

UNCLASSIFIED ONR-CR215-288-3F NL

3 OF 3
ADA
058466



END
DATE
FILMED
11-78
DDC

RESULTS FOR CLASS 1

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
1	0.127E+02	0.188E+06	0.433E+08	0.382E+05	0.432E+09	0.565E+10	0.943E+12	0.180E+13
1	0.135E+02	0.143E+06	0.554E+08	0.389E+05	0.562E+09	0.488E+10	0.592E+12	0.170E+13
1	0.202E+02	0.743E+05	0.130E+09	0.387E+05	0.486E+09	0.448E+10	0.541E+12	0.126E+13
1	0.156E+02	0.124E+06	0.694E+08	0.349E+05	0.394E+09	0.323E+10	0.120E+13	0.404E+13
1	0.155E+02	0.110E+06	0.551E+08	0.381E+05	0.373E+09	0.338E+10	0.103E+13	0.383E+13
1	0.197E+02	0.988E+05	0.144E+09	0.378E+05	0.568E+09	0.361E+10	0.642E+12	0.116E+13
1	0.149E+02	0.274E+06	0.408E+08	0.401E+05	0.437E+09	0.724E+10	0.205E+13	0.269E+13
1	0.163E+02	0.302E+06	0.397E+08	0.435E+05	0.511E+09	0.777E+10	0.209E+13	0.258E+13
1	0.311E+02	0.557E+06	0.148E+10	0.113E+07	0.201E+11	0.951E+11	0.997E+13	0.989E+14
1	0.455E+02	0.113E+07	0.110E+10	0.384E+07	0.860E+10	0.340E+11	0.114E+14	0.981E+14
1	0.292E+02	0.502E+06	0.133E+10	0.378E+07	0.157E+11	0.569E+11	0.956E+13	0.103E+15
1	0.323E+02	0.416E+06	0.715E+09	0.433E+05	0.108E+11	0.349E+11	0.211E+14	0.466E+14
1	0.353E+02	0.613E+06	0.104E+10	0.688E+05	0.158E+11	0.608E+11	0.280E+14	0.621E+14
1	0.310E+02	0.587E+06	0.148E+10	0.427E+07	0.183E+11	0.639E+11	0.112E+14	0.126E+15
1	0.395E+02	0.723E+06	0.785E+09	0.283E+07	0.563E+10	0.238E+11	0.733E+13	0.542E+14
1	0.329E+02	0.531E+06	0.155E+10	0.116E+07	0.211E+11	0.102E+12	0.102E+14	0.104E+15
1	0.246E+02	0.107E+07	0.156E+10	0.109E+06	0.148E+11	0.721E+10	0.217E+14	0.295E+14
1	0.267E+02	0.151E+07	0.173E+10	0.146E+06	0.981E+10	0.750E+10	0.248E+14	0.391E+14
1	0.320E+02	0.202E+07	0.275E+10	0.223E+06	0.275E+11	0.134E+11	0.411E+14	0.538E+14
1	0.349E+02	0.202E+07	0.176E+10	0.782E+05	0.117E+11	0.198E+11	0.543E+14	0.488E+14
1	0.308E+02	0.164E+07	0.148E+10	0.618E+05	0.973E+10	0.152E+11	0.440E+14	0.402E+14
1	0.262E+02	0.168E+07	0.228E+10	0.182E+06	0.235E+11	0.112E+11	0.348E+14	0.446E+14
1	0.362E+02	0.193E+07	0.225E+10	0.180E+06	0.130E+11	0.925E+10	0.321E+14	0.503E+14
1	0.253E+02	0.870E+06	0.128E+10	0.151E+06	0.127E+11	0.617E+10	0.178E+14	0.239E+14
1	0.300E+02	0.439E+06	0.541E+09	0.163E+06	0.168E+11	0.669E+11	0.987E+13	0.942E+14
1	0.340E+02	0.714E+06	0.787E+09	0.630E+05	0.184E+11	0.155E+12	0.102E+14	0.362E+14
1	0.336E+02	0.457E+06	0.249E+10	0.104E+06	0.227E+11	0.216E+12	0.128E+14	0.380E+14
1	0.430E+02	0.548E+06	0.809E+09	0.279E+07	0.210E+11	0.739E+11	0.141E+14	0.214E+15
1	0.416E+02	0.269E+06	0.573E+09	0.109E+07	0.126E+11	0.511E+11	0.870E+13	0.130E+15
1	0.354E+02	0.512E+06	0.262E+10	0.171E+06	0.244E+11	0.239E+12	0.131E+14	0.414E+14
1	0.293E+02	0.750E+06	0.846E+09	0.599E+05	0.197E+11	0.164E+12	0.117E+14	0.394E+14
1	0.280E+02	0.603E+06	0.688E+09	0.300E+06	0.217E+11	0.835E+11	0.131E+14	0.125E+15
1	0.136E+02	0.564E+04	0.958E+07	0.134E+05	0.548E+08	0.108E+10	0.750E+12	0.744E+12
1	0.152E+02	0.613E+04	0.226E+08	0.139E+05	0.154E+09	0.202E+10	0.173E+13	0.165E+13
1	0.289E+02	0.100E+05	0.419E+08	0.191E+05	0.249E+09	0.315E+09	0.100E+13	0.920E+12
1	0.219E+02	0.101E+05	0.301E+08	0.109E+05	0.120E+09	0.441E+09	0.479E+12	0.105E+13
1	0.258E+02	0.188E+05	0.189E+08	0.132E+05	0.111E+09	0.720E+09	0.703E+12	0.127E+13
1	0.293E+02	0.906E+04	0.308E+08	0.149E+05	0.373E+09	0.315E+09	0.130E+13	0.106E+13
1	0.131E+02	0.508E+04	0.136E+08	0.124E+05	0.914E+08	0.141E+10	0.103E+13	0.980E+12
1	0.126E+02	0.680E+04	0.156E+08	0.127E+05	0.101E+09	0.153E+10	0.128E+13	0.127E+13
1	0.148E+02	0.863E+05	0.126E+09	0.926E+05	0.612E+09	0.169E+10	0.121E+13	0.645E+13
1	0.217E+02	0.782E+05	0.200E+09	0.597E+05	0.240E+10	0.416E+10	0.199E+13	0.657E+13
1	0.248E+02	0.102E+06	0.131E+09	0.455E+05	0.809E+09	0.315E+10	0.133E+13	0.667E+13
1	0.222E+02	0.177E+06	0.183E+09	0.519E+05	0.137E+10	0.105E+11	0.701E+13	0.781E+13
1	0.192E+02	0.147E+06	0.144E+09	0.429E+05	0.108E+10	0.872E+10	0.529E+13	0.605E+13
1	0.254E+02	0.666E+05	0.842E+08	0.311E+05	0.525E+09	0.228E+10	0.865E+12	0.479E+13
1	0.152E+02	0.444E+05	0.111E+09	0.440E+05	0.148E+10	0.307E+10	0.180E+13	0.424E+13
1	0.154E+02	0.597E+05	0.890E+08	0.687E+05	0.452E+09	0.134E+10	0.879E+12	0.508E+13
1	0.178E+02	0.521E+05	0.482E+08	0.108E+05	0.115E+10	0.480E+09	0.123E+13	0.155E+13
1	0.183E+02	0.590E+05	0.273E+08	0.850E+04	0.252E+09	0.131E+10	0.437E+12	0.184E+13
1	0.159E+02	0.572E+05	0.644E+08	0.387E+04	0.102E+10	0.648E+09	0.122E+13	0.194E+13
1	0.244E+02	0.682E+05	0.100E+09	0.239E+05	0.540E+09	0.147E+10	0.116E+13	0.109E+13
1	0.211E+02	0.329E+05	0.591E+08	0.228E+05	0.282E+09	0.228E+10	0.301E+12	0.499E+12
1	0.183E+02	0.108E+06	0.137E+09	0.409E+04	0.177E+10	0.948E+09	0.245E+13	0.339E+13
1	0.167E+02	0.460E+05	0.171E+08	0.120E+05	0.321E+09	0.148E+10	0.249E+12	0.171E+13
1	0.190E+02	0.523E+05	0.397E+08	0.139E+05	0.115E+10	0.581E+09	0.119E+13	0.194E+13
1	0.177E+02	0.574E+05	0.635E+08	0.140E+05	0.389E+09	0.134E+10	0.846E+12	0.391E+13
1	0.203E+02	0.383E+05	0.663E+08	0.498E+05	0.113E+10	0.221E+10	0.674E+12	0.354E+13
1	0.165E+02	0.376E+05	0.607E+08	0.330E+05	0.317E+09	0.697E+09	0.601E+12	0.332E+13
1	0.185E+02	0.930E+05	0.339E+08	0.190E+05	0.650E+09	0.750E+10	0.377E+13	0.427E+13
1	0.216E+02	0.154E+06	0.130E+09	0.213E+05	0.104E+10	0.101E+11	0.616E+13	0.680E+13
1	0.188E+02	0.670E+05	0.112E+09	0.403E+05	0.480E+09	0.104E+10	0.110E+13	0.497E+13
1	0.206E+02	0.482E+05	0.112E+09	0.434E+05	0.150E+10	0.200E+10	0.127E+13	0.396E+13
1	0.148E+02	0.453E+05	0.544E+08	0.127E+05	0.347E+09	0.140E+10	0.702E+12	0.358E+13

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 2

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
2	0.801E+02	0.154E+02	0.117E+06	0.956E+03	0.292E+07	0.105E+08	0.283E+10	0.660E+10
2	0.794E+02	0.170E+02	0.121E+06	0.101E+04	0.300E+07	0.920E+07	0.267E+10	0.572E+10
2	0.978E+02	0.335E+02	0.268E+06	0.791E+03	0.544E+07	0.837E+07	0.250E+10	0.560E+10
2	0.619E+02	0.334E+02	0.360E+06	0.270E+04	0.126E+07	0.242E+08	0.386E+10	0.101E+11
2	0.591E+02	0.362E+02	0.515E+06	0.290E+04	0.188E+07	0.184E+08	0.370E+10	0.100E+11
2	0.107E+03	0.261E+02	0.205E+06	0.900E+03	0.458E+07	0.999E+07	0.203E+10	0.485E+10
2	0.808E+02	0.183E+02	0.118E+06	0.103E+04	0.217E+07	0.903E+07	0.325E+10	0.692E+10
2	0.854E+02	0.149E+02	0.136E+06	0.870E+03	0.385E+07	0.100E+08	0.298E+10	0.665E+10
2	0.194E+03	0.411E+02	0.364E+07	0.821E+04	0.675E+08	0.315E+09	0.387E+11	0.680E+12
2	0.166E+03	0.421E+02	0.279E+07	0.775E+04	0.344E+08	0.454E+09	0.216E+11	0.178E+12
2	0.169E+03	0.273E+02	0.600E+07	0.600E+04	0.645E+08	0.627E+09	0.281E+11	0.323E+12
2	0.180E+03	0.245E+02	0.233E+07	0.140E+05	0.680E+08	0.278E+09	0.381E+11	0.480E+12
2	0.186E+03	0.331E+02	0.259E+07	0.102E+05	0.732E+08	0.297E+09	0.396E+11	0.510E+12
2	0.180E+03	0.348E+02	0.749E+07	0.784E+04	0.791E+08	0.763E+09	0.338E+11	0.397E+12
2	0.157E+03	0.266E+02	0.261E+07	0.872E+04	0.253E+08	0.320E+09	0.166E+11	0.164E+12
2	0.210E+03	0.305E+02	0.272E+07	0.889E+04	0.573E+08	0.241E+09	0.300E+11	0.535E+12
2	0.382E+02	0.200E+02	0.114E+06	0.247E+03	0.171E+07	0.336E+07	0.300E+10	0.449E+10
2	0.471E+02	0.217E+02	0.134E+06	0.280E+03	0.571E+06	0.167E+07	0.459E+10	0.773E+10
2	0.848E+02	0.127E+02	0.112E+06	0.396E+03	0.230E+07	0.127E+07	0.288E+10	0.209E+10
2	0.546E+02	0.220E+02	0.141E+06	0.147E+03	0.530E+06	0.357E+07	0.290E+10	0.262E+10
2	0.562E+02	0.321E+02	0.396E+06	0.233E+03	0.213E+07	0.205E+07	0.462E+10	0.606E+10
2	0.857E+02	0.138E+02	0.147E+06	0.377E+03	0.290E+07	0.166E+07	0.395E+10	0.331E+10
2	0.465E+02	0.170E+02	0.832E+05	0.270E+03	0.411E+06	0.149E+07	0.286E+10	0.500E+10
2	0.373E+02	0.114E+02	0.439E+05	0.222E+03	0.676E+06	0.206E+07	0.237E+10	0.344E+10
2	0.275E+03	0.348E+02	0.876E+07	0.134E+05	0.282E+09	0.977E+09	0.324E+12	0.236E+13
2	0.196E+03	0.468E+02	0.107E+08	0.100E+05	0.221E+09	0.187E+10	0.245E+12	0.640E+12
2	0.627E+02	0.375E+02	0.372E+08	0.271E+05	0.270E+09	0.304E+10	0.311E+12	0.633E+12
2	0.263E+03	0.254E+02	0.749E+07	0.309E+04	0.192E+09	0.608E+09	0.138E+12	0.212E+13
2	0.269E+03	0.368E+02	0.928E+07	0.272E+04	0.227E+09	0.750E+09	0.157E+12	0.255E+13
2	0.628E+02	0.360E+02	0.347E+08	0.246E+05	0.252E+09	0.282E+10	0.273E+12	0.580E+12
2	0.206E+03	0.425E+02	0.801E+07	0.122E+05	0.177E+09	0.151E+10	0.181E+12	0.515E+12
2	0.277E+03	0.351E+02	0.821E+07	0.152E+05	0.274E+09	0.977E+09	0.288E+12	0.239E+13
2	0.593E+02	0.171E+02	0.135E+06	0.620E+03	0.610E+07	0.790E+07	0.274E+10	0.557E+10
2	0.981E+02	0.283E+02	0.870E+05	0.663E+03	0.115E+08	0.137E+08	0.410E+10	0.708E+10
2	0.469E+02	0.294E+02	0.589E+06	0.378E+04	0.211E+07	0.381E+08	0.519E+10	0.690E+10
2	0.497E+02	0.304E+02	0.451E+06	0.337E+04	0.183E+07	0.305E+08	0.408E+10	0.598E+10
2	0.984E+02	0.354E+02	0.699E+05	0.740E+03	0.951E+07	0.142E+08	0.341E+10	0.109E+11
2	0.667E+02	0.192E+02	0.104E+06	0.705E+03	0.488E+07	0.757E+07	0.229E+10	0.453E+10
2	0.577E+02	0.235E+02	0.128E+06	0.640E+03	0.497E+07	0.570E+07	0.220E+10	0.443E+10
2	0.644E+02	0.160E+02	0.371E+05	0.410E+03	0.348E+06	0.405E+06	0.478E+09	0.142E+10
2	0.555E+02	0.130E+02	0.497E+05	0.388E+03	0.942E+06	0.338E+06	0.217E+10	0.234E+10
2	0.647E+02	0.129E+02	0.372E+05	0.431E+03	0.227E+06	0.315E+06	0.101E+10	0.224E+10
2	0.856E+02	0.236E+02	0.428E+05	0.129E+03	0.298E+06	0.550E+07	0.400E+10	0.330E+10
2	0.866E+02	0.250E+02	0.468E+05	0.135E+03	0.330E+06	0.602E+07	0.433E+10	0.358E+10
2	0.612E+02	0.989E+01	0.443E+05	0.300E+03	0.289E+06	0.409E+06	0.852E+09	0.213E+10
2	0.570E+02	0.140E+02	0.398E+05	0.423E+03	0.680E+06	0.280E+06	0.148E+10	0.164E+10
2	0.632E+02	0.167E+02	0.406E+05	0.390E+03	0.400E+06	0.425E+06	0.432E+09	0.137E+10
2	0.675E+02	0.165E+02	0.399E+05	0.832E+03	0.463E+06	0.221E+06	0.618E+09	0.763E+09
2	0.809E+02	0.272E+02	0.817E+05	0.967E+03	0.278E+06	0.496E+06	0.861E+09	0.215E+10
2	0.573E+02	0.132E+02	0.402E+05	0.773E+03	0.461E+06	0.416E+06	0.497E+09	0.120E+10
2	0.101E+03	0.160E+02	0.851E+05	0.557E+03	0.533E+06	0.650E+06	0.920E+09	0.930E+09
2	0.936E+02	0.148E+02	0.676E+05	0.531E+03	0.403E+06	0.660E+06	0.632E+09	0.703E+09
2	0.596E+02	0.140E+02	0.443E+05	0.814E+03	0.586E+06	0.471E+06	0.629E+09	0.147E+10
2	0.810E+02	0.212E+02	0.561E+05	0.825E+03	0.174E+06	0.418E+06	0.380E+09	0.135E+10
2	0.669E+02	0.180E+02	0.469E+05	0.827E+03	0.582E+06	0.277E+06	0.859E+09	0.103E+10
2	0.510E+02	0.219E+02	0.997E+05	0.229E+03	0.169E+07	0.654E+06	0.411E+10	0.475E+10
2	0.516E+02	0.212E+02	0.782E+05	0.167E+03	0.353E+06	0.354E+06	0.200E+10	0.417E+10
2	0.490E+02	0.125E+02	0.566E+05	0.139E+03	0.968E+06	0.439E+06	0.231E+10	0.266E+10
2	0.495E+02	0.277E+02	0.169E+06	0.190E+03	0.172E+07	0.655E+06	0.710E+10	0.593E+10
2	0.492E+02	0.253E+02	0.167E+06	0.963E+02	0.168E+07	0.645E+06	0.699E+10	0.589E+10
2	0.457E+02	0.141E+02	0.101E+06	0.115E+03	0.166E+07	0.618E+06	0.445E+10	0.513E+10
2	0.487E+02	0.237E+02	0.982E+05	0.139E+03	0.466E+06	0.413E+06	0.271E+10	0.527E+10
2	0.491E+02	0.194E+02	0.838E+05	0.193E+03	0.141E+07	0.536E+06	0.356E+10	0.409E+10

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 3

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
3	0.681E+02	0.109E+03	0.174E+02	0.5e4E+03	0.125E+05	0.140E+06	0.450E+09	0.352E+09
3	0.685E+02	0.107E+03	0.173E+02	0.5+0E+03	0.158E+05	0.169E+06	0.514E+09	0.406E+09
3	0.695E+02	0.841E+02	0.279E+02	0.6J5E+03	0.285E+05	0.135E+05	0.322E+09	0.229E+09
3	0.548E+02	0.858E+02	0.355E+02	0.673E+03	0.114E+05	0.506E+04	0.205E+09	0.303E+09
3	0.532E+02	0.882E+02	0.303E+02	0.600E+03	0.729E+04	0.326E+04	0.142E+09	0.207E+09
3	0.692E+02	0.837E+02	0.278E+02	0.610E+03	0.262E+05	0.127E+05	0.298E+09	0.213E+09
3	0.675E+02	0.106E+03	0.204E+02	0.535E+03	0.169E+05	0.178E+06	0.539E+09	0.425E+09
3	0.685E+02	0.107E+03	0.192E+02	0.542E+03	0.147E+05	0.157E+06	0.484E+09	0.383E+09
3	0.749E+02	0.101E+03	0.286E+02	0.557E+03	0.107E+05	0.164E+06	0.360E+08	0.367E+09
3	0.676E+02	0.146E+03	0.429E+02	0.653E+03	0.148E+06	0.965E+05	0.735E+08	0.527E+09
3	0.754E+02	0.145E+03	0.366E+02	0.774E+03	0.128E+05	0.142E+06	0.392E+08	0.635E+09
3	0.918E+02	0.262E+03	0.311E+02	0.5+3E+03	0.114E+05	0.181E+06	0.756E+08	0.236E+09
3	0.862E+02	0.252E+03	0.422E+02	0.358E+03	0.138E+05	0.208E+06	0.868E+08	0.257E+09
3	0.742E+02	0.140E+03	0.298E+02	0.7J8E+03	0.112E+05	0.124E+06	0.342E+08	0.545E+09
3	0.703E+02	0.147E+03	0.344E+02	0.7e7E+03	0.126E+06	0.741E+05	0.595E+08	0.413E+09
3	0.766E+02	0.159E+03	0.374E+02	0.6e4E+03	0.136E+05	0.222E+06	0.428E+08	0.450E+09
3	0.498E+02	0.798E+02	0.257E+02	0.833E+02	0.729E+04	0.438E+04	0.947E+08	0.119E+09
3	0.499E+02	0.827E+02	0.268E+02	0.97+E+02	0.488E+04	0.432E+04	0.102E+09	0.178E+09
3	0.571E+02	0.115E+03	0.315E+02	0.78+E+02	0.101E+05	0.494E+04	0.129E+09	0.152E+09
3	0.569E+02	0.115E+03	0.269E+02	0.4e8E+02	0.107E+05	0.523E+04	0.149E+09	0.121E+09
3	0.581E+02	0.113E+03	0.269E+02	0.5e3E+02	0.148E+05	0.643E+04	0.204E+09	0.162E+09
3	0.561E+02	0.121E+03	0.221E+02	0.500E+02	0.656E+04	0.323E+04	0.943E+08	0.109E+09
3	0.481E+02	0.873E+02	0.224E+02	0.5J7E+02	0.351E+04	0.506E+04	0.625E+08	0.122E+09
3	0.498E+02	0.828E+02	0.251E+02	0.722E+02	0.624E+04	0.375E+04	0.871E+08	0.109E+09
3	0.688E+02	0.179E+03	0.354E+02	0.373E+03	0.352E+03	0.700E+04	0.258E+07	0.226E+08
3	0.635E+02	0.142E+03	0.230E+02	0.3e2E+03	0.197E+05	0.813E+04	0.587E+07	0.286E+08
3	0.647E+02	0.145E+03	0.295E+02	0.3J5E+03	0.354E+03	0.576E+04	0.451E+07	0.441E+08
3	0.688E+02	0.215E+03	0.226E+02	0.2e3E+03	0.962E+03	0.108E+05	0.296E+07	0.764E+07
3	0.709E+02	0.225E+03	0.290E+02	0.26+E+03	0.119E+04	0.124E+05	0.357E+07	0.975E+07
3	0.636E+02	0.147E+03	0.232E+02	0.27e+03	0.320E+03	0.623E+04	0.388E+07	0.472E+08
3	0.657E+02	0.146E+03	0.343E+02	0.492E+03	0.193E+05	0.802E+04	0.518E+07	0.216E+08
3	0.672E+02	0.178E+03	0.256E+02	0.2e7E+03	0.225E+03	0.967E+04	0.261E+07	0.272E+08
3	0.499E+02	0.106E+03	0.109E+02	0.43e+02	0.717E+02	0.235E+03	0.350E+06	0.730E+06
3	0.506E+02	0.111E+03	0.141E+02	0.49+E+02	0.109E+03	0.290E+03	0.329E+06	0.890E+06
3	0.541E+02	0.122E+03	0.257E+02	0.43+E+02	0.298E+03	0.693E+03	0.686E+06	0.130E+07
3	0.544E+02	0.145E+03	0.325E+02	0.8J2E+02	0.299E+03	0.215E+04	0.112E+07	0.126E+07
3	0.542E+02	0.136E+03	0.294E+02	0.8J9E+02	0.295E+03	0.219E+04	0.140E+07	0.136E+07
3	0.545E+02	0.129E+03	0.209E+02	0.3++E+02	0.189E+03	0.583E+03	0.489E+06	0.107E+07
3	0.500E+02	0.109E+03	0.102E+02	0.3J3E+02	0.78JE+02	0.192E+03	0.261E+06	0.804E+06
3	0.497E+02	0.106E+03	0.115E+02	0.3J7E+02	0.675E+02	0.229E+03	0.599E+06	0.797E+06
3	0.551E+02	0.130E+03	0.137E+02	0.33+E+02	0.163E+03	0.298E+03	0.619E+04	0.325E+04
4	0.566E+02	0.161E+03	0.216E+02	0.215E+02	0.343E+03	0.183E+03	0.159E+05	0.511E+05
3	0.525E+02	0.114E+03	0.123E+02	0.30+E+02	0.207E+03	0.466E+03	0.329E+05	0.397E+05
3	0.538E+02	0.126E+03	0.150E+02	0.2J7E+02	0.105E+03	0.349E+03	0.648E+04	0.211E+05
3	0.531E+02	0.120E+03	0.139E+02	0.2J0E+02	0.102E+03	0.364E+03	0.453E+04	0.173E+05
3	0.526E+02	0.118E+03	0.136E+02	0.27+E+02	0.201E+03	0.398E+03	0.369E+05	0.350E+05
4	0.568E+02	0.165E+03	0.231E+02	0.217E+02	0.341E+03	0.172E+03	0.105E+05	0.406E+05
3	0.558E+02	0.137E+03	0.153E+02	0.2e6E+02	0.171E+03	0.267E+03	0.862E+04	0.375E+04
3	0.514E+02	0.926E+02	0.146E+02	0.17JE+02	0.123E+04	0.980E+03	0.218E+08	0.243E+08
4	0.511E+02	0.121E+03	0.226E+02	0.1+0E+02	0.686E+03	0.422E+03	0.119E+08	0.200E+08
3	0.506E+02	0.904E+02	0.125E+02	0.166E+02	0.175E+04	0.967E+03	0.212E+08	0.227E+08
4	0.532E+02	0.102E+03	0.187E+02	0.13JE+02	0.343E+04	0.110E+04	0.287E+08	0.212E+08
4	0.535E+02	0.105E+03	0.195E+02	0.14+E+02	0.341E+04	0.125E+04	0.284E+08	0.212E+08
3	0.504E+02	0.877E+02	0.160E+02	0.19+E+02	0.152E+04	0.105E+04	0.184E+08	0.199E+08
4	0.512E+02	0.115E+03	0.195E+02	0.9J3E+01	0.792E+03	0.474E+03	0.147E+08	0.242E+08
3	0.519E+02	0.953E+02	0.124E+02	0.1J1E+02	0.890E+03	0.881E+03	0.174E+08	0.194E+08
3	0.580E+02	0.138E+03	0.247E+02	0.4+1E+02	0.997E+03	0.134E+04	0.141E+08	0.354E+08
3	0.578E+02	0.149E+03	0.216E+02	0.33JE+02	0.593E+03	0.706E+04	0.109E+08	0.277E+08
3	0.503E+02	0.109E+03	0.286E+02	0.4++E+02	0.847E+03	0.131E+04	0.120E+08	0.337E+08
3	0.626E+02	0.169E+03	0.289E+02	0.6J1E+02	0.998E+03	0.201E+05	0.531E+08	0.412E+08
3	0.627E+02	0.176E+03	0.295E+02	0.9J7E+02	0.477E+03	0.154E+05	0.173E+08	0.240E+08
3	0.504E+02	0.106E+03	0.283E+02	0.4+4E+02	0.562E+03	0.349E+04	0.578E+07	0.316E+08
3	0.571E+02	0.146E+03	0.207E+02	0.2e0E+02	0.572E+03	0.480E+04	0.143E+08	0.254E+08
3	0.581E+02	0.142E+03	0.271E+02	0.3e5E+02	0.127E+04	0.614E+03	0.217E+08	0.393E+08

--NUMBER OF CORRECT CLASSIF. : 58
NUMBER OF ERRONEOUS CLASSIF. : 6
PERCENTAGE OF SUCCESS : 90.62500

RESULTS FOR CLASS 4

CLASSIF.: DISTANCES TO CLASSES:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
4	0.477E+02	0.584E+03	0.929E+06	0.1[illegible]E+02	0.163E+08	0.116E+09	0.182E+12	0.150E+12
4	0.518E+02	0.487E+03	0.812E+06	0.1[illegible]E+02	0.147E+08	0.107E+09	0.167E+12	0.138E+12
4	0.837E+02	0.544E+03	0.230E+07	0.2[illegible]E+02	0.235E+08	0.304E+08	0.148E+12	0.121E+12
4	0.390E+02	0.533E+03	0.179E+07	0.25[illegible]E+02	0.126E+08	0.714E+07	0.570E+11	0.841E+11
4	0.386E+02	0.502E+03	0.171E+07	0.25[illegible]E+02	0.121E+08	0.684E+07	0.551E+11	0.814E+11
4	0.859E+02	0.478E+03	0.211E+07	0.26[illegible]E+02	0.219E+08	0.277E+08	0.140E+12	0.114E+12
4	0.475E+02	0.500E+03	0.808E+06	0.1[illegible]E+02	0.145E+08	0.105E+09	0.163E+12	0.134E+12
4	0.435E+02	0.528E+03	0.852E+06	0.1[illegible]E+02	0.149E+08	0.107E+09	0.165E+12	0.136E+12
4	0.103E+03	0.736E+03	0.204E+08	0.3[illegible]E+02	0.259E+09	0.281E+10	0.129E+12	0.121E+13
4	0.933E+02	0.884E+04	[illegible].157E+09	0.31[illegible]E+02	0.337E+09	0.927E+09	0.303E+12	0.252E+13
4	0.204E+03	0.901E+03	0.961E+07	0.37[illegible]E+02	0.242E+09	0.102E+10	0.214E+12	0.300E+13
4	0.189E+03	0.854E+03	0.930E+07	0.37[illegible]E+02	0.230E+09	0.222E+10	0.192E+12	0.586E+12
4	0.180E+03	0.693E+03	0.852E+07	0.29[illegible]E+02	0.211E+09	0.213E+10	0.165E+12	0.522E+12
4	0.199E+03	0.856E+03	0.895E+07	0.36[illegible]E+02	0.232E+09	0.979E+09	0.205E+12	0.287E+13
4	0.885E+02	0.848E+04	0.154E+08	0.30[illegible]E+02	0.327E+09	0.906E+09	0.293E+12	0.243E+13
4	0.112E+03	0.885E+03	0.199E+08	0.42[illegible]E+02	0.258E+09	0.277E+10	0.130E+12	0.126E+13
4	0.517E+02	0.249E+04	0.933E+07	0.10[illegible]E+02	0.106E+09	0.397E+08	0.260E+12	0.328E+12
4	0.356E+02	0.173E+04	0.677E+07	0.10[illegible]E+02	0.453E+08	0.263E+08	0.176E+12	0.261E+12
4	0.692E+02	0.171E+04	0.775E+07	0.21[illegible]E+02	0.924E+08	0.335E+08	0.233E+12	0.291E+12
4	0.657E+02	0.306E+04	0.933E+07	0.29[illegible]E+02	0.829E+08	0.123E+09	0.482E+12	0.402E+12
4	0.651E+02	0.356E+04	0.103E+08	0.27[illegible]E+02	0.903E+08	0.135E+09	0.522E+12	0.437E+12
4	0.669E+02	0.128E+04	0.544E+07	0.23[illegible]E+02	0.647E+08	0.235E+08	0.164E+12	0.204E+12
4	0.358E+02	0.195E+04	0.7[illegible]0E+07	0.19[illegible]E+02	0.524E+08	0.307E+08	0.206E+12	0.305E+12
4	0.556E+02	0.192E+04	0.814E+07	0.20[illegible]E+02	0.947E+08	0.347E+08	0.237E+12	0.299E+12
4	0.150E+03	0.444E+04	0.161E+08	0.17[illegible]E+02	0.586E+09	0.227E+10	0.440E+12	0.573E+13
4	0.138E+03	0.478E+04	0.199E+08	0.3[illegible]E+02	0.446E+09	0.545E+10	0.245E+12	0.130E+13
4	0.101E+03	0.227E+04	0.606E+08	0.41[illegible]E+02	0.590E+09	0.682E+10	0.321E+12	0.154E+13
4	0.162E+03	0.506E+04	0.186E+08	0.19[illegible]E+02	0.677E+09	0.262E+10	0.509E+12	0.664E+13
4	0.165E+03	0.380E+04	0.164E+08	0.22[illegible]E+02	0.586E+09	0.229E+10	0.450E+12	0.594E+13
4	0.979E+02	0.219E+04	0.596E+08	0.37[illegible]E+02	0.589E+09	0.681E+10	0.315E+12	0.158E+13
4	0.175E+03	0.437E+04	0.241E+08	0.39[illegible]E+02	0.536E+09	0.663E+10	0.298E+12	0.159E+13
4	0.172E+03	0.429E+04	0.179E+08	0.10[illegible]E+02	0.642E+09	0.251E+10	0.490E+12	0.645E+13
4	0.534E+02	0.116E+03	0.273E+02	0.100E+02	0.729E+02	0.146E+03	0.736E+06	0.192E+07
4	0.537E+02	0.123E+03	0.228E+02	0.10[illegible]E+02	0.665E+02	0.116E+03	0.640E+06	0.216E+07
4	0.509E+02	0.105E+03	0.178E+02	0.14[illegible]E+02	0.149E+03	0.225E+03	0.607E+06	0.219E+07
4	0.544E+02	0.134E+03	0.302E+02	0.21[illegible]E+02	0.275E+03	0.239E+04	0.317E+07	0.271E+07
4	0.545E+02	0.139E+03	0.311E+02	0.20[illegible]E+02	0.237E+03	0.361E+04	0.506E+07	0.425E+07
4	0.509E+02	0.103E+03	0.161E+02	0.15[illegible]E+02	0.142E+03	0.286E+03	0.948E+06	0.330E+07
4	0.533E+02	0.114E+03	0.183E+02	0.12[illegible]E+02	0.823E+02	0.126E+03	0.644E+06	0.223E+07
4	0.540E+02	0.116E+03	0.256E+02	0.12[illegible]E+02	0.700E+02	0.130E+03	0.757E+06	0.254E+07
4	0.525E+02	0.268E+03	0.143E+07	0.2[illegible]E+02	0.187E+08	0.623E+07	0.520E+11	0.639E+11
4	0.459E+02	0.193E+03	0.115E+07	0.22[illegible]E+02	0.813E+07	0.462E+07	0.385E+11	0.586E+11
4	0.526E+02	0.235E+03	0.117E+07	0.20[illegible]E+02	0.153E+08	0.504E+07	0.442E+11	0.541E+11
4	0.566E+02	0.272E+03	0.163E+07	0.23[illegible]E+02	0.166E+08	0.195E+08	0.183E+12	0.851E+11
4	0.578E+02	0.255E+03	0.155E+07	0.30[illegible]E+02	0.161E+08	0.185E+08	0.993E+11	0.823E+11
4	0.534E+02	0.234E+03	0.109E+07	0.24[illegible]E+02	0.139E+08	0.465E+07	0.397E+11	0.487E+11
4	0.450E+02	0.179E+03	0.114E+07	0.23[illegible]E+02	0.775E+07	0.441E+07	0.371E+11	0.567E+11
4	0.508E+02	0.216E+03	0.109E+07	0.23[illegible]E+02	0.145E+08	0.473E+07	0.411E+11	0.502E+11
4	0.710E+02	0.504E+04	0.173E+09	0.17[illegible]E+02	0.193E+09	0.738E+08	0.466E+12	0.589E+12
4	0.436E+02	0.552E+04	0.161E+08	0.12[illegible]E+02	0.107E+09	0.621E+08	0.375E+12	0.557E+12
4	0.584E+02	0.427E+04	0.133E+08	0.15[illegible]E+02	0.146E+09	0.564E+08	0.344E+12	0.438E+12
4	0.856E+02	0.627E+04	0.171E+08	0.24[illegible]E+02	0.149E+09	0.228E+09	0.863E+12	0.722E+12
4	0.851E+02	0.717E+04	0.180E+08	0.35[illegible]E+02	0.156E+09	0.242E+09	0.896E+12	0.751E+12
4	0.595E+02	0.392E+04	0.142E+08	0.12[illegible]E+02	0.160E+09	0.606E+08	0.388E+12	0.491E+12
4	0.451E+02	0.391E+04	0.133E+08	0.15[illegible]E+02	0.888E+08	0.515E+08	0.330E+12	0.490E+12
4	0.695E+02	0.341E+04	0.123E+08	0.18[illegible]E+02	0.138E+09	0.522E+08	0.340E+12	0.428E+12
4	0.410E+02	0.597E+04	0.180E+08	0.17[illegible]E+02	0.120E+09	0.700E+08	0.430E+12	0.637E+12
4	0.670E+02	0.881E+04	0.271E+08	0.21[illegible]E+02	0.295E+09	0.115E+09	0.686E+12	0.874E+12
4	0.478E+02	0.677E+04	0.203E+08	0.13[illegible]E+02	0.135E+09	0.786E+08	0.482[illegible]E+12	0.715E+12
4	0.657E+02	0.113E+05	0.897E+07	0.31[illegible]E+02	0.122E+09	0.801E+09	0.113E+13	0.951E+12
4	0.630E+02	0.136E+05	0.111E+08	0.35[illegible]E+02	0.152E+09	0.101E+10	0.142E+13	0.119E+13
4	0.479E+02	0.693E+04	0.200E+08	0.15[illegible]E+02	0.133E+09	0.776E+08	0.470E+12	0.598E+12
4	0.652E+02	0.884E+04	0.279E+08	0.22[illegible]E+02	0.305E+09	0.119E+09	0.713E+12	0.907E+12
4	0.392E+02	0.733E+04	0.207E+08	0.23[illegible]E+02	0.137E+09	0.802E+08	0.481E+12	0.712E+12

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 5

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
5	0.499E+02	0.926E+02	0.170E+03	0.287E+02	0.145E+02	0.526E+03	0.168E+07	0.133E+07
5	0.497E+02	0.892E+02	0.161E+03	0.362E+02	0.167E+02	0.576E+03	0.165E+07	0.133E+07
5	0.565E+02	0.122E+03	0.351E+03	0.492E+02	0.285E+02	0.618E+03	0.993E+06	0.460E+06
5	0.493E+02	0.952E+02	0.268E+03	0.314E+02	0.242E+02	0.364E+03	0.642E+06	0.748E+06
5	0.491E+02	0.938E+02	0.346E+03	0.443E+02	0.256E+02	0.349E+03	0.925E+06	0.104E+07
5	0.562E+02	0.123E+03	0.365E+03	0.495E+02	0.273E+02	0.630E+03	0.786E+06	0.339E+06
5	0.494E+02	0.883E+02	0.181E+03	0.340E+02	0.173E+02	0.535E+03	0.276E+07	0.214E+07
5	0.499E+02	0.908E+02	0.156E+03	0.268E+02	0.156E+02	0.570E+03	0.254E+07	0.196E+07
5	0.648E+02	0.165E+03	0.191E+03	0.398E+03	0.351E+02	0.239E+04	0.249E+07	0.226E+08
5	0.561E+02	0.806E+02	0.204E+03	0.316E+03	0.306E+02	0.294E+04	0.173E+07	0.147E+08
5	0.685E+02	0.219E+03	0.199E+03	0.300E+03	0.323E+02	0.175E+04	0.300E+07	0.185E+08
5	0.804E+02	0.222E+03	0.125E+03	0.103E+03	0.349E+02	0.207E+04	0.788E+07	0.144E+08
5	0.789E+02	0.218E+03	0.108E+03	0.103E+03	0.350E+02	0.149E+04	0.563E+07	0.991E+07
5	0.696E+02	0.225E+03	0.199E+03	0.317E+03	0.316E+02	0.167E+04	0.303E+07	0.185E+08
5	0.571E+02	0.793E+02	0.243E+03	0.347E+03	0.431E+02	0.383E+04	0.228E+07	0.197E+08
5	0.629E+02	0.164E+03	0.179E+03	0.349E+03	0.298E+02	0.139E+04	0.151E+07	0.138E+08
5	0.542E+02	0.153E+03	0.221E+03	0.153E+03	0.185E+02	0.138E+03	0.283E+07	0.267E+07
5	0.554E+02	0.110E+03	0.201E+03	0.131E+03	0.220E+02	0.141E+03	0.226E+07	0.372E+07
5	0.587E+02	0.201E+03	0.242E+03	0.109E+03	0.245E+02	0.197E+03	0.237E+07	0.224E+07
5	0.606E+02	0.171E+03	0.197E+03	0.641E+02	0.268E+02	0.127E+03	0.336E+07	0.212E+07
5	0.618E+02	0.172E+03	0.248E+03	0.729E+02	0.272E+02	0.174E+03	0.502E+07	0.315E+07
5	0.589E+02	0.199E+03	0.235E+03	0.118E+03	0.233E+02	0.203E+03	0.266E+07	0.253E+07
5	0.547E+02	0.130E+03	0.312E+03	0.961E+02	0.236E+02	0.141E+03	0.136E+07	0.206E+07
5	0.537E+02	0.149E+03	0.261E+03	0.153E+03	0.260E+02	0.167E+03	0.328E+07	0.313E+07
5	0.720E+02	0.262E+03	0.162E+03	0.350E+03	0.246E+02	0.215E+05	0.633E+07	0.132E+09
5	0.745E+02	0.202E+03	0.317E+03	0.322E+03	0.298E+02	0.406E+05	0.303E+07	0.251E+08
5	0.638E+02	0.145E+03	0.420E+03	0.372E+03	0.370E+02	0.388E+05	0.536E+07	0.338E+08
5	0.714E+02	0.257E+03	0.149E+03	0.330E+03	0.183E+02	0.238E+05	0.672E+07	0.141E+09
5	0.716E+02	0.256E+03	0.153E+03	0.398E+03	0.296E+02	0.255E+05	0.744E+07	0.158E+09
5	0.655E+02	0.146E+03	0.489E+03	0.457E+03	0.401E+02	0.409E+05	0.613E+07	0.366E+08
5	0.760E+02	0.208E+03	0.341E+03	0.353E+03	0.354E+02	0.485E+05	0.359E+07	0.272E+08
5	0.720E+02	0.202E+03	0.166E+03	0.331E+03	0.259E+02	0.234E+05	0.652E+07	0.131E+09
5	0.614E+02	0.158E+03	0.158E+03	0.133E+03	0.151E+02	0.179E+03	0.148E+05	0.343E+05
5	0.607E+02	0.160E+03	0.132E+03	0.110E+03	0.165E+02	0.171E+03	0.132E+05	0.456E+05
5	0.581E+02	0.147E+03	0.908E+02	0.107E+03	0.313E+02	0.287E+03	0.619E+05	0.930E+05
5	0.576E+02	0.170E+03	0.155E+03	0.130E+03	0.264E+02	0.616E+03	0.103E+06	0.654E+05
5	0.579E+02	0.170E+03	0.160E+03	0.145E+03	0.283E+02	0.617E+03	0.498E+05	0.397E+05
5	0.592E+02	0.145E+03	0.112E+03	0.144E+03	0.280E+02	0.313E+03	0.440E+05	0.858E+05
5	0.620E+02	0.159E+03	0.176E+03	0.153E+03	0.150E+02	0.175E+03	0.138E+05	0.377E+05
5	0.623E+02	0.158E+03	0.122E+03	0.101E+03	0.166E+02	0.176E+03	0.137E+05	0.306E+05
5	0.578E+02	0.988E+02	0.130E+03	0.174E+03	0.194E+02	0.187E+03	0.615E+07	0.869E+07
5	0.568E+02	0.141E+03	0.232E+03	0.102E+03	0.251E+02	0.135E+03	0.840E+07	0.797E+07
5	0.569E+02	0.886E+02	0.156E+03	0.180E+03	0.137E+02	0.131E+03	0.575E+07	0.788E+07
5	0.568E+02	0.102E+03	0.118E+03	0.952E+02	0.172E+02	0.198E+04	0.191E+08	0.136E+08
5	0.587E+02	0.995E+02	0.144E+03	0.931E+02	0.378E+02	0.309E+04	0.277E+08	0.197E+08
5	0.561E+02	0.883E+02	0.149E+03	0.174E+03	0.141E+02	0.111E+03	0.567E+07	0.774E+07
5	0.560E+02	0.135E+03	0.182E+03	0.131E+03	0.247E+02	0.138E+03	0.689E+07	0.659E+07
5	0.579E+02	0.101E+03	0.122E+03	0.171E+03	0.172E+02	0.203E+03	0.511E+07	0.726E+07
5	0.571E+02	0.104E+03	0.294E+03	0.173E+03	0.125E+02	0.964E+02	0.460E+07	0.644E+07
5	0.557E+02	0.148E+03	0.225E+03	0.122E+03	0.168E+02	0.775E+02	0.488E+07	0.442E+07
5	0.557E+02	0.104E+03	0.264E+03	0.130E+03	0.192E+02	0.137E+03	0.343E+07	0.480E+07
5	0.588E+02	0.110E+03	0.985E+02	0.500E+02	0.254E+02	0.786E+03	0.112E+08	0.784E+07
5	0.583E+02	0.105E+03	0.858E+02	0.548E+02	0.202E+02	0.145E+04	0.150E+08	0.106E+08
5	0.559E+02	0.104E+03	0.303E+03	0.140E+03	0.212E+02	0.179E+03	0.355E+07	0.504E+07
5	0.556E+02	0.158E+03	0.229E+03	0.118E+03	0.136E+02	0.822E+02	0.375E+07	0.336E+07
5	0.569E+02	0.102E+03	0.290E+03	0.174E+03	0.135E+02	0.657E+02	0.445E+07	0.605E+07
5	0.605E+02	0.160E+03	0.243E+03	0.244E+03	0.196E+02	0.548E+02	0.401E+07	0.382E+07
5	0.672E+02	0.142E+03	0.947E+02	0.225E+03	0.147E+02	0.458E+02	0.320E+07	0.477E+07
5	0.600E+02	0.181E+03	0.245E+03	0.234E+03	0.163E+02	0.484E+02	0.389E+07	0.368E+07
5	0.651E+02	0.199E+03	0.116E+03	0.193E+03	0.288E+02	0.365E+02	0.808E+07	0.510E+07
5	0.652E+02	0.195E+03	0.134E+03	0.197E+03	0.294E+02	0.430E+02	0.893E+07	0.560E+07
5	0.601E+02	0.179E+03	0.238E+03	0.235E+03	0.161E+02	0.459E+02	0.350E+07	0.330E+07
5	0.680E+02	0.135E+03	0.879E+02	0.200E+03	0.211E+02	0.433E+02	0.417E+07	0.609E+07
5	0.612E+02	0.176E+03	0.273E+03	0.279E+03	0.272E+02	0.523E+02	0.467E+07	0.450E+07

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.000000

RESULTS FOR CLASS 6

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
6	0.589E+02	0.128E+03	0.564E+02	0.632E+02	0.126E+03	0.189E+02	0.241E+07	0.166E+07
6	0.587E+02	0.130E+03	0.559E+02	0.670E+02	0.141E+03	0.171E+02	0.176E+07	0.120E+07
6	0.608E+02	0.171E+03	0.677E+02	0.5[illegible]E+02	0.202E+03	0.275E+02	0.100E+07	0.580E+06
6	0.567E+02	0.123E+03	0.864E+02	0.810E+02	0.725E+02	0.236E+02	0.478E+06	0.622E+06
6	0.565E+02	0.129E+03	0.712E+02	0.647E+02	0.651E+02	0.216E+02	0.315E+06	0.409E+06
6	0.612E+02	0.172E+03	0.937E+02	0.500E+02	0.242E+03	0.292E+02	0.101E+07	0.568E+06
6	0.584E+02	0.131E+03	0.555E+02	0.621E+02	0.136E+03	0.175E+02	0.164E+07	0.112E+07
6	0.587E+02	0.127E+03	0.552E+02	0.665E+02	0.117E+03	0.265E+02	0.270E+07	0.187E+07
6	0.570E+02	0.168E+03	0.181E+03	0.114E+03	0.139E+03	0.295E+02	0.230E+06	0.220E+07
6	0.596E+02	0.137E+03	0.180E+03	0.121E+03	0.111E+04	0.308E+02	0.329E+06	0.188E+07
6	0.581E+02	0.182E+03	0.164E+03	0.803E+02	0.978E+02	0.304E+02	0.891E+05	0.216E+07
6	0.651E+02	0.178E+03	0.206E+03	0.114E+03	0.778E+03	0.374E+02	0.325E+06	0.152E+07
6	0.637E+02	0.173E+03	0.223E+03	0.118E+03	0.607E+03	0.311E+02	0.172E+06	0.869E+06
6	0.584E+02	0.180E+03	0.166E+03	0.950E+02	0.933E+02	0.276E+02	0.982E+05	0.233E+07
6	0.588E+02	0.139E+03	0.188E+03	0.102E+03	0.130E+04	0.338E+02	0.361E+06	0.203E+07
6	0.568E+02	0.164E+03	0.200E+03	0.118E+03	0.176E+03	0.333E+02	0.271E+06	0.259E+07
6	0.564E+02	0.163E+03	0.569E+02	0.423E+02	0.252E+03	0.183E+02	0.661E+05	0.187E+07
6	0.598E+02	0.150E+03	0.107E+03	0.101E+03	0.476E+03	0.332E+02	0.597E+05	0.561E+06
6	0.554E+02	0.141E+03	0.120E+03	0.714E+02	0.340E+03	0.329E+02	0.170E+06	0.109E+07
6	0.572E+02	0.168E+03	0.706E+02	0.456E+02	0.237E+03	0.174E+02	0.706E+05	0.196E+07
6	0.580E+02	0.178E+03	0.841E+02	0.673E+02	0.343E+03	0.477E+02	0.168E+06	0.448E+07
6	0.556E+02	0.141E+03	0.128E+03	0.781E+02	0.334E+03	0.313E+02	0.151E+06	0.965E+06
6	0.599E+02	0.152E+03	0.108E+03	0.979E+02	0.466E+03	0.286E+02	0.541E+05	0.494E+06
6	0.569E+02	0.164E+03	0.648E+02	0.427E+02	0.230E+03	0.151E+02	0.694E+05	0.189E+07
6	0.592E+02	0.192E+03	0.214E+03	0.151E+03	0.191E+02	0.189E+02	0.136E+07	0.118E+07
6	0.637E+02	0.164E+03	0.817E+02	0.120E+03	0.318E+02	0.253E+02	0.481E+06	0.658E+06
6	0.591E+02	0.193E+03	0.198E+03	0.160E+03	0.292E+02	0.166E+02	0.134E+07	0.117E+07
6	0.616E+02	0.198E+03	0.116E+03	0.115E+03	0.849E+02	0.226E+02	0.156E+07	0.880E+06
6	0.615E+02	0.206E+03	0.144E+03	0.936E+02	0.109E+03	0.313E+02	0.126E+07	0.688E+06
6	0.590E+02	0.193E+03	0.175E+03	0.133E+03	0.370E+02	0.186E+02	0.734E+06	0.617E+06
5	0.636E+02	0.164E+03	0.766E+02	0.131E+03	0.295E+02	0.296E+02	0.492E+06	0.666E+06
6	0.589E+02	0.196E+03	0.202E+03	0.130E+03	0.222E+02	0.163E+02	0.908E+06	0.757E+06
6	0.717E+02	0.165E+03	0.348E+03	0.502E+03	0.527E+02	0.942E+01	0.520E+02	0.116E+03
6	0.711E+02	0.165E+03	0.322E+03	0.502E+03	0.544E+02	0.961E+01	0.455E+02	0.102E+03
6	0.706E+02	0.164E+03	0.297E+03	0.509E+03	0.538E+02	0.143E+02	0.542E+02	0.865E+02
6	0.677E+02	0.163E+03	0.392E+03	0.504E+03	0.804E+02	0.305E+02	0.176E+03	0.890E+02
6	0.671E+02	0.161E+03	0.373E+03	0.571E+03	0.828E+02	0.286E+02	0.183E+03	0.127E+03
6	0.708E+02	0.165E+03	0.303E+03	0.551E+03	0.527E+02	0.147E+02	0.492E+02	0.765E+02
6	0.709E+02	0.166E+03	0.316E+03	0.543E+03	0.559E+02	0.108E+02	0.449E+02	0.114E+03
6	0.713E+02	0.165E+03	0.328E+03	0.561E+03	0.540E+02	0.986E+01	0.519E+02	0.126E+03
6	0.622E+02	0.178E+03	0.270E+03	0.336E+03	0.263E+02	0.143E+02	0.145E+07	0.125E+07
6	0.686E+02	0.124E+03	0.941E+02	0.336E+03	0.528E+02	0.280E+02	0.132E+07	0.180E+07
6	0.630E+02	0.179E+03	0.267E+03	0.302E+03	0.263E+02	0.152E+02	0.142E+07	0.125E+07
6	0.665E+02	0.188E+03	0.109E+03	0.225E+03	0.688E+02	0.301E+02	0.216E+07	0.121E+07
6	0.675E+02	0.184E+03	0.127E+03	0.290E+03	0.429E+02	0.224E+02	0.271E+07	0.156E+07
6	0.629E+02	0.177E+03	0.280E+03	0.300E+03	0.208E+02	0.130E+02	0.187E+07	0.167E+07
6	0.686E+02	0.128E+03	0.934E+02	0.330E+03	0.409E+02	0.226E+02	0.103E+07	0.143E+07
6	0.627E+02	0.174E+03	0.284E+03	0.301E+03	0.229E+02	0.118E+02	0.145E+07	0.127E+07
6	0.611E+02	0.166E+03	0.317E+03	0.335E+03	0.105E+03	0.169E+02	0.801E+07	0.774E+07
6	0.674E+02	0.974E+02	0.152E+03	0.372E+03	0.101E+03	0.299E+02	0.775E+07	0.110E+08
6	0.608E+02	0.155E+03	0.295E+03	0.350E+03	0.111E+03	0.258E+02	0.953E+07	0.933E+07
6	0.708E+02	0.176E+03	0.117E+03	0.190E+03	0.377E+03	0.290E+02	0.172E+08	0.113E+08
6	0.712E+02	0.176E+03	0.131E+03	0.219E+03	0.384E+03	0.317E+02	0.167E+08	0.111E+08
6	0.606E+02	0.159E+03	0.283E+03	0.335E+03	0.647E+02	0.220E+02	0.857E+07	0.825E+07
6	0.663E+02	0.101E+03	0.145E+03	0.349E+03	0.699E+02	0.272E+02	0.604E+07	0.884E+07
6	0.618E+02	0.162E+03	0.352E+03	0.376E+03	0.202E+03	0.324E+02	0.980E+07	0.979E+07
6	0.609E+02	0.170E+03	0.286E+03	0.310E+03	0.447E+02	0.160E+02	0.662E+07	0.623E+07
6	0.669E+02	0.104E+03	0.138E+03	0.322E+03	0.114E+03	0.237E+02	0.526E+07	0.709E+07
6	0.597E+02	0.163E+03	0.274E+03	0.306E+03	0.273E+02	0.212E+02	0.610E+07	0.559E+07
6	0.707E+02	0.191E+03	0.136E+03	0.159E+03	0.133E+03	0.284E+02	0.821E+07	0.477E+07
6	0.730E+02	0.186E+03	0.145E+03	0.157E+03	0.190E+03	0.350E+02	0.143E+08	0.867E+07
6	0.600E+02	0.162E+03	0.283E+03	0.310E+03	0.343E+02	0.189E+02	0.677E+07	0.628E+07
6	0.675E+02	0.103E+03	0.147E+03	0.340E+03	0.122E+03	0.305E+02	0.677E+07	0.917E+07
6	0.616E+02	0.168E+03	0.332E+03	0.302E+03	0.808E+02	0.227E+02	0.948E+07	0.906E+07

--NUMBER OF CORRECT CLASSIF. : 63
NUMBER OF ERRONEOUS CLASSIF. : 1
PERCENTAGE OF SUCCESS : 98.43750

RESULTS FOR CLASS 7

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
7	0.616E+02	0.143E+03	0.155E+03	0.305E+03	0.816E+02	0.113E+03	0.278E+02	0.417E+02
7	0.616E+02	0.144E+03	0.154E+03	0.302E+03	0.782E+02	0.111E+03	0.159E+02	0.286E+02
7	0.637E+02	0.160E+03	0.201E+03	0.359E+03	0.794E+02	0.911E+02	0.209E+02	0.432E+02
7	0.655E+02	0.147E+03	0.144E+03	0.306E+03	0.541E+02	0.429E+02	0.172E+02	0.493E+02
7	0.657E+02	0.147E+03	0.140E+03	0.300E+03	0.608E+02	0.497E+02	0.187E+02	0.562E+02
7	0.636E+02	0.161E+03	0.200E+03	0.340E+03	0.834E+02	0.105E+03	0.221E+02	0.448E+02
7	0.618E+02	0.144E+03	0.161E+03	0.307E+03	0.815E+02	0.114E+03	0.154E+02	0.373E+02
7	0.621E+02	0.143E+03	0.164E+03	0.305E+03	0.795E+02	0.110E+03	0.156E+02	0.411E+02
7	0.630E+02	0.162E+03	0.213E+03	0.300E+03	0.773E+02	0.859E+02	0.296E+02	0.105E+03
7	0.639E+02	0.148E+03	0.232E+03	0.296E+03	0.920E+02	0.448E+02	0.303E+02	0.715E+02
7	0.629E+02	0.162E+03	0.213E+03	0.302E+03	0.778E+02	0.127E+03	0.306E+02	0.158E+03
7	0.642E+02	0.161E+03	0.279E+03	0.317E+03	0.778E+02	0.113E+03	0.350E+02	0.194E+03
7	0.633E+02	0.159E+03	0.266E+03	0.296E+03	0.734E+02	0.972E+02	0.294E+02	0.784E+02
7	0.636E+02	0.160E+03	0.230E+03	0.333E+03	0.660E+02	0.103E+03	0.291E+02	0.695E+02
7	0.645E+02	0.145E+03	0.261E+03	0.333E+03	0.931E+02	0.559E+02	0.334E+02	0.209E+03
7	0.636E+02	0.162E+03	0.234E+03	0.330E+03	0.704E+02	0.966E+02	0.325E+02	0.173E+03
7	0.666E+02	0.168E+03	0.180E+03	0.445E+03	0.104E+03	0.136E+03	0.281E+02	0.926E+02
7	0.689E+02	0.145E+03	0.208E+03	0.403E+03	0.867E+02	0.808E+02	0.456E+02	0.158E+03
7	0.669E+02	0.175E+03	0.177E+03	0.435E+03	0.116E+03	0.130E+03	0.261E+02	0.448E+02
7	0.663E+02	0.168E+03	0.280E+03	0.437E+03	0.890E+02	0.113E+03	0.446E+02	0.745E+02
7	0.661E+02	0.166E+03	0.279E+03	0.440E+03	0.827E+02	0.966E+02	0.186E+02	0.506E+02
7	0.672E+02	0.174E+03	0.185E+03	0.444E+03	0.107E+03	0.115E+03	0.214E+02	0.405E+02
7	0.684E+02	0.147E+03	0.196E+03	0.440E+03	0.716E+02	0.591E+02	0.228E+02	0.509E+02
7	0.673E+02	0.169E+03	0.196E+03	0.404E+03	0.941E+02	0.114E+03	0.295E+02	0.665E+02
7	0.708E+02	0.159E+03	0.415E+03	0.601E+03	0.521E+02	0.256E+02	0.120E+02	0.510E+02
7	0.708E+02	0.158E+03	0.408E+03	0.603E+03	0.520E+02	0.231E+02	0.109E+02	0.397E+02
7	0.706E+02	0.158E+03	0.393E+03	0.600E+03	0.483E+02	0.216E+02	0.116E+02	0.319E+02
6	0.709E+02	0.158E+03	0.380E+03	0.606E+03	0.457E+02	0.138E+02	0.286E+02	0.208E+02
6	0.709E+02	0.158E+03	0.385E+03	0.607E+03	0.454E+02	0.129E+02	0.267E+02	0.233E+02
7	0.709E+02	0.158E+03	0.403E+03	0.605E+03	0.509E+02	0.214E+02	0.129E+02	0.330E+02
7	0.709E+02	0.158E+03	0.400E+03	0.606E+03	0.482E+02	0.199E+02	0.847E+01	0.306E+02
7	0.705E+02	0.158E+03	0.397E+03	0.605E+03	0.499E+02	0.245E+02	0.100E+02	0.484E+02
7	0.641E+02	0.180E+03	0.247E+03	0.206E+03	0.782E+02	0.293E+03	0.306E+02	0.117E+05
7	0.662E+02	0.166E+03	0.310E+03	0.270E+03	0.210E+03	0.337E+03	0.327E+02	0.258E+04
7	0.599E+02	0.137E+03	0.199E+03	0.258E+03	0.125E+03	0.516E+03	0.394E+02	0.120E+04
7	0.599E+02	0.165E+03	0.192E+03	0.211E+03	0.116E+03	0.235E+03	0.493E+02	0.473E+04
7	0.613E+02	0.172E+03	0.250E+03	0.245E+03	0.116E+03	0.226E+03	0.571E+02	0.173E+05
7	0.592E+02	0.136E+03	0.185E+03	0.233E+03	0.151E+03	0.553E+03	0.472E+02	0.258E+04
7	0.661E+02	0.168E+03	0.298E+03	0.250E+03	0.215E+03	0.343E+03	0.356E+02	0.274E+04
7	0.643E+02	0.179E+03	0.245E+03	0.210E+03	0.743E+02	0.284E+03	0.369E+02	0.926E+04
6	0.700E+02	0.163E+03	0.281E+03	0.513E+03	0.506E+02	0.135E+02	0.324E+02	0.122E+03
6	0.698E+02	0.165E+03	0.257E+03	0.502E+03	0.548E+02	0.131E+02	0.203E+02	0.117E+03
6	0.701E+02	0.164E+03	0.275E+03	0.504E+03	0.572E+02	0.127E+02	0.207E+02	0.149E+03
7	0.661E+02	0.159E+03	0.339E+03	0.524E+03	0.926E+02	0.511E+02	0.296E+02	0.403E+03
7	0.664E+02	0.159E+03	0.345E+03	0.535E+03	0.968E+02	0.560E+02	0.236E+02	0.329E+03
6	0.700E+02	0.164E+03	0.279E+03	0.503E+03	0.564E+02	0.118E+02	0.248E+02	0.131E+03
6	0.704E+02	0.165E+03	0.282E+03	0.535E+03	0.489E+02	0.120E+02	0.188E+02	0.963E+02
6	0.693E+02	0.166E+03	0.254E+03	0.481E+03	0.563E+02	0.155E+02	0.234E+02	0.127E+03
7	0.680E+02	0.170E+03	0.217E+03	0.407E+03	0.793E+02	0.614E+02	0.214E+02	0.834E+02
7	0.688E+02	0.157E+03	0.204E+03	0.440E+03	0.574E+02	0.307E+02	0.153E+02	0.108E+03
7	0.671E+02	0.164E+03	0.192E+03	0.400E+03	0.778E+02	0.903E+02	0.177E+02	0.992E+02
7	0.663E+02	0.166E+03	0.285E+03	0.400E+03	0.812E+02	0.781E+02	0.178E+02	0.302E+02
7	0.658E+02	0.167E+03	0.274E+03	0.431E+03	0.858E+02	0.808E+02	0.188E+02	0.323E+02
7	0.675E+02	0.165E+03	0.199E+03	0.472E+03	0.759E+02	0.824E+02	0.166E+02	0.953E+02
7	0.688E+02	0.153E+03	0.210E+03	0.442E+03	0.605E+02	0.473E+02	0.185E+02	0.859E+02
7	0.679E+02	0.171E+03	0.209E+03	0.454E+03	0.868E+02	0.658E+02	0.202E+02	0.737E+02
7	0.683E+02	0.163E+03	0.229E+03	0.407E+03	0.597E+02	0.292E+02	0.108E+02	0.690E+02
7	0.679E+02	0.157E+03	0.198E+03	0.434E+03	0.490E+02	0.167E+02	0.157E+02	0.745E+02
7	0.678E+02	0.163E+03	0.213E+03	0.430E+03	0.601E+02	0.313E+02	0.118E+02	0.969E+02
7	0.656E+02	0.160E+03	0.278E+03	0.470E+03	0.722E+02	0.518E+02	0.105E+02	0.385E+02
7	0.658E+02	0.160E+03	0.285E+03	0.479E+03	0.690E+02	0.432E+02	0.226E+02	0.538E+02
7	0.683E+02	0.163E+03	0.231E+03	0.478E+03	0.560E+02	0.284E+02	0.121E+02	0.776E+02
7	0.682E+02	0.156E+03	0.210E+03	0.432E+03	0.469E+02	0.164E+02	0.117E+02	0.531E+02
7	0.682E+02	0.162E+03	0.227E+03	0.400E+03	0.596E+02	0.317E+02	0.105E+02	0.708E+02

--NUMBER OF CORRECT CLASSIF. : 56
NUMBER OF ERRONEOUS CLASSIF. : 8
PERCENTAGE OF SUCCESS : 87.50000

RESULTS FOR CLASS 8

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
8	0.624E+02	0.148E+03	0.182E+03	0.397E+03	0.903E+02	0.880E+02	0.522E+02	0.202E+02
8	0.629E+02	0.149E+03	0.195E+03	0.410E+03	0.852E+02	0.764E+02	0.592E+02	0.232E+02
8	0.636E+02	0.162E+03	0.211E+03	0.300E+03	0.863E+02	0.856E+02	0.523E+02	0.246E+02
8	0.668E+02	0.153E+03	0.149E+03	0.373E+03	0.593E+02	0.404E+02	0.351E+02	0.307E+02
8	0.666E+02	0.153E+03	0.140E+03	0.365E+03	0.571E+02	0.377E+02	0.416E+02	0.251E+02
8	0.643E+02	0.162E+03	0.229E+03	0.395E+03	0.836E+02	0.870E+02	0.557E+02	0.255E+02
8	0.626E+02	0.147E+03	0.187E+03	0.409E+03	0.916E+02	0.897E+02	0.603E+02	0.154E+02
8	0.625E+02	0.148E+03	0.184E+03	0.406E+03	0.909E+02	0.866E+02	0.688E+02	0.216E+02
7	0.647E+02	0.163E+03	0.266E+03	0.364E+03	0.607E+02	0.625E+02	0.239E+02	0.388E+02
8	0.660E+02	0.150E+03	0.302E+03	0.377E+03	0.754E+02	0.446E+02	0.375E+02	0.361E+02
8	0.642E+02	0.158E+03	0.249E+03	0.350E+03	0.572E+02	0.733E+02	0.478E+02	0.277E+02
8	0.656E+02	0.161E+03	0.333E+03	0.371E+03	0.662E+02	0.901E+02	0.377E+02	0.300E+02
8	0.654E+02	0.162E+03	0.328E+03	0.375E+03	0.609E+02	0.788E+02	0.316E+02	0.281E+02
8	0.639E+02	0.157E+03	0.246E+03	0.347E+03	0.613E+02	0.797E+02	0.808E+02	0.343E+02
7	0.651E+02	0.150E+03	0.286E+03	0.344E+03	0.787E+02	0.419E+02	0.352E+02	0.376E+02
7	0.642E+02	0.162E+03	0.249E+03	0.333E+03	0.685E+02	0.570E+02	0.249E+02	0.356E+02
8	0.677E+02	0.175E+03	0.189E+03	0.439E+03	0.123E+03	0.114E+03	0.338E+02	0.267E+02
7	0.696E+02	0.156E+03	0.215E+03	0.450E+03	0.781E+02	0.535E+02	0.242E+02	0.253E+02
8	0.677E+02	0.176E+03	0.186E+03	0.443E+03	0.117E+03	0.116E+03	0.295E+02	0.208E+02
8	0.653E+02	0.171E+03	0.297E+03	0.435E+03	0.977E+02	0.969E+02	0.881E+02	0.259E+02
8	0.659E+02	0.171E+03	0.313E+03	0.450E+03	0.961E+02	0.100E+03	0.956E+02	0.242E+02
8	0.680E+02	0.175E+03	0.191E+03	0.447E+03	0.113E+03	0.962E+02	0.390E+02	0.227E+02
8	0.695E+02	0.155E+03	0.215E+03	0.440E+03	0.807E+02	0.605E+02	0.302E+02	0.201E+02
8	0.679E+02	0.174E+03	0.191E+03	0.441E+03	0.125E+03	0.109E+03	0.431E+02	0.215E+02
8	0.689E+02	0.168E+03	0.401E+03	0.405E+03	0.306E+02	0.690E+02	0.918E+02	0.306E+02
8	0.683E+02	0.161E+03	0.405E+03	0.403E+03	0.769E+02	0.942E+02	0.398E+02	0.280E+02
8	0.662E+02	0.149E+03	0.349E+03	0.396E+03	0.494E+02	0.121E+03	0.117E+03	0.467E+02
8	0.671E+02	0.163E+03	0.363E+03	0.405E+03	0.396E+02	0.471E+02	0.278E+02	0.209E+02
8	0.663E+02	0.163E+03	0.340E+03	0.447E+03	0.421E+02	0.618E+02	0.349E+02	0.333E+02
8	0.661E+02	0.148E+03	0.339E+03	0.491E+03	0.501E+02	0.109E+03	0.178E+03	0.373E+02
7	0.697E+02	0.163E+03	0.425E+03	0.510E+03	0.743E+02	0.760E+02	0.358E+02	0.450E+02
5	0.677E+02	0.167E+03	0.383E+03	0.435E+03	0.304E+02	0.828E+02	0.121E+03	0.363E+02
8	0.736E+02	0.162E+03	0.509E+03	0.760E+03	0.566E+02	0.194E+02	0.217E+02	0.105E+02
8	0.739E+02	0.162E+03	0.512E+03	0.774E+03	0.554E+02	0.190E+02	0.227E+02	0.994E+01
8	0.738E+02	0.163E+03	0.515E+03	0.770E+03	0.552E+02	0.250E+02	0.319E+02	0.138E+02
8	0.740E+02	0.163E+03	0.523E+03	0.778E+03	0.553E+02	0.219E+02	0.372E+02	0.121E+02
8	0.737E+02	0.162E+03	0.509E+03	0.764E+03	0.538E+02	0.192E+02	0.363E+02	0.139E+02
8	0.741E+02	0.163E+03	0.516E+03	0.772E+03	0.525E+02	0.254E+02	0.329E+02	0.217E+02
8	0.736E+02	0.162E+03	0.503E+03	0.760E+03	0.540E+02	0.175E+02	0.225E+02	0.924E+01
8	0.738E+02	0.162E+03	0.513E+03	0.775E+03	0.565E+02	0.197E+02	0.222E+02	0.117E+02
8	0.702E+02	0.176E+03	0.267E+03	0.540E+03	0.783E+02	0.695E+02	0.305E+02	0.226E+02
8	0.718E+02	0.156E+03	0.279E+03	0.537E+03	0.870E+02	0.730E+02	0.487E+02	0.261E+02
8	0.702E+02	0.176E+03	0.262E+03	0.547E+03	0.808E+02	0.707E+02	0.261E+02	0.116E+02
8	0.684E+02	0.174E+03	0.382E+03	0.531E+03	0.981E+02	0.121E+03	0.432E+02	0.191E+02
8	0.676E+02	0.175E+03	0.360E+03	0.517E+03	0.105E+03	0.132E+03	0.376E+02	0.215E+02
8	0.706E+02	0.176E+03	0.277E+03	0.500E+03	0.736E+02	0.621E+02	0.279E+02	0.119E+02
8	0.725E+02	0.154E+03	0.303E+03	0.571E+03	0.899E+02	0.824E+02	0.483E+02	0.155E+02
8	0.708E+02	0.176E+03	0.241E+03	0.501E+03	0.715E+02	0.659E+02	0.323E+02	0.158E+02
8	0.691E+02	0.170E+03	0.260E+03	0.529E+03	0.514E+02	0.433E+02	0.176E+03	0.311E+02
8	0.708E+02	0.153E+03	0.265E+03	0.532E+03	0.546E+02	0.325E+02	0.914E+02	0.233E+02
8	0.692E+02	0.169E+03	0.259E+03	0.529E+03	0.487E+02	0.376E+02	0.804E+02	0.234E+02
8	0.683E+02	0.168E+03	0.339E+03	0.544E+03	0.803E+02	0.667E+02	0.332E+03	0.446E+02
8	0.684E+02	0.166E+03	0.352E+03	0.504E+03	0.766E+02	0.543E+02	0.982E+02	0.210E+02
8	0.697E+02	0.169E+03	0.278E+03	0.531E+03	0.461E+02	0.320E+02	0.154E+03	0.254E+02
8	0.710E+02	0.154E+03	0.273E+03	0.541E+03	0.519E+02	0.300E+02	0.106E+03	0.287E+02
8	0.697E+02	0.169E+03	0.278E+03	0.543E+03	0.462E+02	0.291E+02	0.551E+02	0.228E+02
8	0.724E+02	0.156E+03	0.301E+03	0.506E+03	0.779E+02	0.675E+02	0.583E+02	0.181E+02
8	0.700E+02	0.175E+03	0.256E+03	0.550E+03	0.804E+02	0.587E+02	0.315E+02	0.195E+02
8	0.718E+02	0.157E+03	0.279E+03	0.547E+03	0.691E+02	0.443E+02	0.403E+02	0.149E+02
8	0.664E+02	0.153E+03	0.328E+03	0.595E+03	0.122E+03	0.578E+02	0.207E+03	0.294E+02
8	0.658E+02	0.154E+03	0.309E+03	0.509E+03	0.126E+03	0.633E+02	0.169E+03	0.240E+02
8	0.719E+02	0.156E+03	0.273E+03	0.545E+03	0.783E+02	0.604E+02	0.529E+02	0.158E+02
8	0.699E+02	0.176E+03	0.260E+03	0.539E+03	0.798E+02	0.699E+02	0.273E+02	0.128E+02
8	0.718E+02	0.157E+03	0.274E+03	0.545E+03	0.721E+02	0.554E+02	0.470E+02	0.149E+02

--NUMBER OF CORRECT CLASSIF. : 58
NUMBER OF ERRONEOUS CLASSIF. : 6
PERCENTAGE OF SUCCESS : 90.625UU

RESULTS FOR CLASS 1

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
1	0.158E+02	0.385E+06	0.852E+08	0.920E+05	0.238E+10	0.848E+10	0.278E+13	0.873E+13
1	0.215E+02	0.315E+06	0.152E+09	0.119E+06	0.261E+10	0.101E+11	0.187E+13	0.147E+14
1	0.315E+02	0.570E+06	0.468E+09	0.257E+06	0.463E+10	0.424E+11	0.458E+13	0.777E+13
1	0.265E+02	0.200E+06	0.107E+09	0.119E+06	0.166E+10	0.103E+11	0.251E+13	0.109E+14
1	0.247E+02	0.176E+06	0.953E+08	0.121E+06	0.170E+10	0.104E+11	0.231E+13	0.967E+13
1	0.267E+02	0.603E+06	0.465E+09	0.356E+06	0.571E+10	0.392E+11	0.430E+13	0.899E+13
1	0.171E+02	0.599E+06	0.128E+09	0.976E+05	0.320E+10	0.145E+11	0.675E+13	0.408E+13
1	0.238E+02	0.552E+06	0.954E+08	0.101E+06	0.241E+10	0.105E+11	0.562E+13	0.895E+13
1	0.422E+02	0.346E+07	0.120E+11	0.232E+07	0.199E+12	0.375E+12	0.209E+15	0.206E+15
1	0.615E+02	0.520E+07	0.429E+10	0.171E+08	0.337E+12	0.231E+13	0.228E+15	0.705E+15
1	0.339E+02	0.399E+07	0.554E+10	0.136E+08	0.725E+11	0.730E+12	0.105E+15	0.427E+15
1	0.382E+02	0.316E+07	0.361E+10	0.273E+07	0.121E+12	0.648E+12	0.593E+14	0.154E+15
1	0.444E+02	0.477E+07	0.589E+10	0.397E+07	0.163E+12	0.101E+13	0.739E+14	0.202E+15
1	0.359E+02	0.470E+07	0.672E+10	0.176E+08	0.915E+11	0.946E+12	0.122E+15	0.561E+15
1	0.604E+02	0.303E+07	0.239E+10	0.115E+08	0.224E+12	0.138E+13	0.161E+15	0.375E+15
1	0.425E+02	0.342E+07	0.125E+11	0.220E+07	0.207E+12	0.397E+12	0.217E+15	0.216E+15
1	0.363E+02	0.269E+07	0.257E+10	0.205E+07	0.311E+11	0.135E+12	0.721E+14	0.600E+14
1	0.392E+02	0.296E+07	0.454E+10	0.150E+07	0.519E+11	0.236E+11	0.852E+14	0.854E+14
1	0.364E+02	0.352E+07	0.721E+10	0.115E+07	0.102E+12	0.329E+11	0.111E+15	0.103E+15
1	0.399E+02	0.920E+07	0.103E+11	0.178E+07	0.448E+11	0.233E+12	0.194E+15	0.224E+15
1	0.356E+02	0.837E+07	0.957E+10	0.190E+07	0.407E+11	0.274E+12	0.174E+15	0.247E+15
1	0.295E+02	0.292E+07	0.614E+10	0.970E+06	0.868E+11	0.275E+11	0.986E+14	0.925E+14
1	0.452E+02	0.340E+07	0.540E+10	0.105E+07	0.613E+11	0.311E+11	0.961E+14	0.975E+14
1	0.302E+02	0.231E+07	0.226E+10	0.192E+07	0.273E+11	0.126E+12	0.684E+14	0.557E+14
1	0.466E+02	0.244E+07	0.120E+11	0.729E+06	0.101E+12	0.125E+13	0.108E+15	0.104E+16
1	0.393E+02	0.334E+07	0.112E+11	0.400E+06	0.783E+11	0.960E+12	0.598E+14	0.875E+14
1	0.491E+02	0.951E+07	0.191E+11	0.709E+06	0.115E+12	0.115E+13	0.622E+14	0.108E+15
1	0.576E+02	0.385E+07	0.127E+11	0.834E+07	0.215E+12	0.244E+13	0.126E+15	0.149E+16
1	0.550E+02	0.219E+07	0.747E+10	0.500E+07	0.114E+12	0.137E+13	0.728E+14	0.867E+15
1	0.493E+02	0.107E+08	0.209E+11	0.850E+06	0.112E+12	0.120E+13	0.713E+14	0.107E+15
1	0.368E+02	0.357E+07	0.113E+11	0.473E+06	0.823E+11	0.989E+12	0.586E+14	0.916E+14
1	0.492E+02	0.325E+07	0.150E+11	0.140E+07	0.139E+12	0.165E+13	0.136E+15	0.131E+16
1	0.231E+02	0.110E+05	0.219E+08	0.795E+05	0.404E+09	0.430E+10	0.205E+13	0.133E+13
1	0.209E+02	0.176E+05	0.849E+08	0.951E+05	0.610E+09	0.102E+11	0.476E+13	0.301E+13
1	0.336E+02	0.319E+05	0.122E+09	0.699E+05	0.263E+10	0.633E+10	0.278E+13	0.201E+13
1	0.323E+02	0.444E+05	0.343E+08	0.587E+05	0.914E+09	0.460E+10	0.155E+13	0.198E+13
1	0.395E+02	0.440E+05	0.302E+08	0.600E+05	0.921E+09	0.895E+10	0.177E+13	0.275E+13
1	0.316E+02	0.205E+05	0.154E+09	0.553E+05	0.185E+10	0.102E+10	0.313E+13	0.151E+13
1	0.175E+02	0.126E+05	0.435E+08	0.105E+06	0.460E+09	0.703E+10	0.311E+13	0.172E+13
1	0.144E+02	0.146E+05	0.522E+08	0.104E+06	0.533E+09	0.910E+10	0.357E+13	0.240E+13
1	0.288E+02	0.232E+06	0.316E+09	0.912E+06	0.110E+11	0.115E+12	0.204E+14	0.368E+14
1	0.335E+02	0.221E+06	0.879E+09	0.187E+06	0.180E+11	0.159E+11	0.175E+14	0.148E+14
1	0.278E+02	0.225E+06	0.696E+09	0.298E+06	0.891E+10	0.188E+11	0.163E+14	0.128E+14
1	0.348E+02	0.115E+07	0.175E+10	0.346E+07	0.138E+11	0.259E+12	0.500E+14	0.138E+15
1	0.292E+02	0.957E+06	0.143E+10	0.298E+07	0.121E+11	0.221E+12	0.400E+14	0.115E+15
1	0.279E+02	0.149E+06	0.471E+09	0.207E+06	0.607E+10	0.137E+11	0.118E+14	0.917E+13
1	0.249E+02	0.137E+06	0.558E+09	0.125E+06	0.119E+11	0.114E+11	0.114E+14	0.957E+13
1	0.246E+02	0.158E+06	0.235E+09	0.684E+06	0.914E+10	0.921E+11	0.162E+14	0.298E+14
1	0.252E+02	0.139E+06	0.125E+09	0.972E+05	0.429E+10	0.762E+10	0.230E+13	0.456E+13
1	0.296E+02	0.119E+06	0.572E+08	0.449E+05	0.191E+10	0.357E+10	0.162E+13	0.238E+13
1	0.277E+02	0.981E+05	0.137E+09	0.379E+05	0.233E+10	0.289E+10	0.201E+13	0.292E+13
1	0.302E+02	0.316E+06	0.362E+09	0.670E+06	0.753E+10	0.361E+11	0.349E+13	0.126E+14
1	0.272E+02	0.184E+06	0.254E+09	0.774E+06	0.634E+10	0.521E+11	0.405E+13	0.179E+14
1	0.298E+02	0.172E+06	0.275E+09	0.425E+05	0.422E+10	0.375E+10	0.389E+13	0.484E+13
1	0.250E+02	0.956E+05	0.639E+08	0.522E+05	0.260E+10	0.365E+10	0.155E+13	0.234E+13
1	0.305E+02	0.138E+06	0.135E+09	0.139E+06	0.556E+10	0.161E+11	0.352E+13	0.806E+13
1	0.236E+02	0.116E+06	0.230E+09	0.674E+05	0.256E+10	0.129E+11	0.517E+13	0.699E+13
1	0.307E+02	0.134E+06	0.177E+09	0.165E+06	0.419E+10	0.172E+11	0.302E+13	0.154E+14
1	0.252E+02	0.842E+05	0.149E+09	0.241E+06	0.634E+10	0.604E+11	0.922E+13	0.200E+14
1	0.343E+02	0.547E+06	0.664E+09	0.242E+06	0.189E+10	0.615E+11	0.860E+13	0.144E+14
1	0.350E+02	0.640E+06	0.975E+09	0.384E+06	0.378E+10	0.906E+11	0.138E+14	0.239E+14
1	0.224E+02	0.148E+06	0.229E+09	0.358E+06	0.855E+10	0.849E+11	0.131E+14	0.274E+14
1	0.274E+02	0.155E+06	0.239E+09	0.192E+06	0.480E+10	0.201E+11	0.448E+13	0.147E+14
1	0.221E+02	0.934E+05	0.238E+09	0.630E+05	0.282E+10	0.124E+11	0.571E+13	0.667E+13

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 2

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
2	0.193E+03	0.170E+02	0.106E+07	0.211E+04	0.221E+08	0.876E+08	0.118E+12	0.111E+12
2	0.193E+03	0.216E+02	0.914E+06	0.257E+04	0.201E+08	0.787E+08	0.110E+12	0.991E+11
2	0.317E+03	0.436E+02	0.372E+07	0.251E+04	0.743E+08	0.408E+09	0.241E+12	0.174E+12
2	0.107E+03	0.453E+02	0.456E+07	0.624E+04	0.399E+08	0.761E+09	0.847E+11	0.713E+12
2	0.112E+03	0.427E+02	0.474E+07	0.584E+04	0.386E+08	0.675E+09	0.893E+11	0.673E+12
2	0.356E+03	0.341E+02	0.292E+07	0.214E+04	0.568E+08	0.307E+09	0.176E+12	0.135E+12
2	0.190E+03	0.236E+02	0.873E+06	0.220E+04	0.185E+08	0.600E+08	0.101E+12	0.823E+11
2	0.209E+03	0.207E+02	0.124E+07	0.197E+04	0.250E+08	0.109E+09	0.132E+12	0.120E+12
2	0.712E+03	0.522E+02	0.254E+08	0.167E+05	0.252E+09	0.298E+10	0.228E+12	0.314E+13
2	0.110E+04	0.561E+02	0.377E+08	0.258E+05	0.369E+09	0.361E+10	0.369E+12	0.402E+12
2	0.901E+03	0.306E+02	0.449E+08	0.125E+05	0.339E+09	0.258E+10	0.429E+12	0.577E+12
2	0.499E+03	0.400E+02	0.433E+08	0.248E+05	0.229E+09	0.332E+10	0.341E+12	0.447E+13
2	0.486E+03	0.483E+02	0.464E+08	0.274E+05	0.248E+09	0.336E+10	0.354E+12	0.476E+13
2	0.903E+03	0.382E+02	0.558E+08	0.141E+05	0.423E+09	0.321E+10	0.520E+12	0.718E+12
2	0.110E+04	0.546E+02	0.307E+08	0.303E+05	0.357E+09	0.299E+10	0.321E+12	0.391E+12
2	0.743E+03	0.380E+02	0.254E+08	0.130E+05	0.201E+09	0.255E+10	0.185E+12	0.296E+13
2	0.921E+02	0.330E+02	0.235E+06	0.940E+03	0.598E+07	0.140E+08	0.314E+11	0.287E+11
2	0.137E+03	0.276E+02	0.397E+06	0.646E+03	0.673E+07	0.165E+08	0.283E+11	0.203E+11
2	0.279E+03	0.249E+02	0.673E+06	0.102E+04	0.154E+08	0.228E+08	0.416E+11	0.190E+11
2	0.159E+03	0.271E+02	0.649E+06	0.811E+03	0.618E+07	0.726E+08	0.139E+11	0.908E+11
2	0.188E+03	0.396E+02	0.185E+07	0.112E+04	0.132E+08	0.832E+08	0.343E+11	0.138E+12
2	0.285E+03	0.235E+02	0.806E+06	0.101E+04	0.187E+08	0.208E+08	0.457E+11	0.212E+11
2	0.124E+03	0.205E+02	0.232E+06	0.499E+03	0.390E+07	0.108E+08	0.216E+11	0.140E+11
2	0.843E+02	0.252E+02	0.106E+06	0.795E+03	0.308E+07	0.552E+07	0.197E+11	0.184E+11
2	0.849E+03	0.549E+02	0.841E+08	0.201E+05	0.162E+10	0.114E+11	0.131E+13	0.942E+13
2	0.903E+03	0.570E+02	0.772E+08	0.329E+05	0.685E+09	0.623E+10	0.124E+13	0.291E+13
2	0.649E+03	0.508E+02	0.136E+09	0.227E+06	0.254E+10	0.136E+11	0.250E+13	0.683E+13
2	0.661E+03	0.441E+02	0.495E+08	0.668E+04	0.137E+10	0.987E+10	0.821E+12	0.771E+13
2	0.655E+03	0.480E+02	0.598E+08	0.637E+04	0.162E+10	0.122E+11	0.958E+12	0.939E+13
2	0.607E+03	0.489E+02	0.129E+09	0.188E+06	0.236E+10	0.124E+11	0.205E+13	0.562E+13
2	0.955E+03	0.541E+02	0.624E+08	0.334E+05	0.530E+09	0.482E+10	0.981E+12	0.272E+13
2	0.823E+03	0.486E+02	0.973E+08	0.235E+05	0.151E+10	0.130E+11	0.133E+13	0.114E+14
2	0.295E+03	0.174E+02	0.392E+07	0.253E+04	0.745E+08	0.222E+09	0.321E+12	0.250E+12
2	0.310E+03	0.166E+02	0.447E+07	0.252E+04	0.831E+08	0.248E+09	0.352E+12	0.289E+12
2	0.260E+03	0.335E+02	0.559E+07	0.622E+04	0.172E+09	0.735E+09	0.466E+12	0.375E+12
2	0.212E+03	0.414E+02	0.146E+08	0.149E+05	0.145E+09	0.232E+10	0.258E+12	0.188E+13
2	0.238E+03	0.344E+02	0.112E+08	0.137E+05	0.114E+09	0.180E+10	0.210E+12	0.154E+13
2	0.263E+03	0.436E+02	0.546E+07	0.731E+04	0.178E+09	0.952E+09	0.516E+12	0.438E+12
2	0.333E+03	0.258E+02	0.420E+07	0.257E+04	0.726E+08	0.221E+09	0.305E+12	0.254E+12
2	0.269E+03	0.310E+02	0.318E+07	0.206E+04	0.604E+08	0.185E+09	0.264E+12	0.199E+12
2	0.152E+03	0.277E+02	0.699E+05	0.785E+03	0.119E+07	0.250E+07	0.215E+10	0.405E+10
2	0.134E+03	0.246E+02	0.832E+05	0.614E+03	0.212E+07	0.111E+07	0.329E+10	0.290E+10
2	0.239E+03	0.227E+02	0.581E+05	0.726E+03	0.128E+07	0.221E+07	0.243E+10	0.314E+10
2	0.301E+03	0.273E+02	0.908E+05	0.475E+03	0.131E+07	0.189E+08	0.123E+11	0.653E+10
2	0.304E+03	0.288E+02	0.991E+05	0.472E+03	0.138E+07	0.200E+08	0.132E+11	0.700E+10
2	0.226E+03	0.189E+02	0.762E+05	0.640E+03	0.175E+07	0.243E+07	0.237E+10	0.306E+10
2	0.139E+03	0.273E+02	0.611E+05	0.670E+03	0.161E+07	0.703E+06	0.215E+10	0.198E+10
2	0.148E+03	0.275E+02	0.762E+05	0.791E+03	0.138E+07	0.276E+07	0.207E+10	0.416E+10
2	0.211E+03	0.252E+02	0.855E+05	0.155E+04	0.846E+06	0.794E+06	0.164E+10	0.235E+10
2	0.256E+03	0.324E+02	0.115E+06	0.188E+04	0.150E+07	0.666E+07	0.299E+10	0.489E+10
2	0.149E+03	0.252E+02	0.109E+06	0.140E+04	0.134E+07	0.429E+07	0.201E+10	0.454E+10
2	0.259E+03	0.323E+02	0.245E+06	0.155E+04	0.240E+07	0.506E+07	0.256E+10	0.619E+10
2	0.239E+03	0.262E+02	0.205E+06	0.114E+04	0.173E+07	0.664E+07	0.223E+10	0.729E+10
2	0.158E+03	0.235E+02	0.122E+06	0.155E+04	0.185E+07	0.591E+07	0.289E+10	0.575E+10
2	0.243E+03	0.257E+02	0.817E+05	0.143E+04	0.121E+07	0.374E+07	0.159E+10	0.334E+10
2	0.213E+03	0.278E+02	0.105E+06	0.157E+04	0.104E+07	0.104E+07	0.194E+10	0.271E+10
2	0.105E+03	0.262E+02	0.197E+06	0.647E+03	0.375E+07	0.262E+07	0.628E+10	0.635E+10
2	0.190E+03	0.257E+02	0.985E+05	0.384E+03	0.164E+07	0.129E+07	0.509E+10	0.644E+10
2	0.120E+03	0.255E+02	0.153E+06	0.470E+03	0.404E+07	0.110E+07	0.545E+10	0.448E+10
2	0.155E+03	0.294E+02	0.511E+06	0.514E+03	0.537E+07	0.286E+07	0.185E+11	0.151E+11
2	0.156E+03	0.264E+02	0.525E+06	0.405E+03	0.612E+07	0.331E+07	0.186E+11	0.167E+11
2	0.109E+03	0.209E+02	0.241E+06	0.340E+03	0.608E+07	0.162E+07	0.922E+10	0.803E+10
2	0.177E+03	0.302E+02	0.128E+06	0.315E+03	0.203E+07	0.175E+07	0.645E+10	0.798E+10
2	0.963E+02	0.245E+02	0.166E+06	0.595E+03	0.312E+07	0.216E+07	0.537E+10	0.535E+10

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 3

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
3	0.173E+03	0.137E+03	0.386E+02	0.803E+03	0.359E+05	0.262E+06	0.117E+10	0.580E+09
3	0.170E+03	0.139E+03	0.298E+02	0.822E+03	0.479E+05	0.329E+06	0.131E+10	0.708E+09
3	0.175E+03	0.223E+03	0.311E+02	0.107E+04	0.149E+06	0.540E+05	0.755E+09	0.350E+09
3	0.147E+03	0.159E+03	0.446E+02	0.109E+04	0.308E+05	0.622E+05	0.450E+09	0.447E+09
3	0.139E+03	0.160E+03	0.377E+02	0.943E+03	0.209E+05	0.402E+05	0.318E+09	0.312E+09
3	0.175E+03	0.216E+03	0.340E+02	0.103E+04	0.148E+06	0.590E+05	0.701E+09	0.344E+09
3	0.158E+03	0.139E+03	0.490E+02	0.874E+03	0.475E+05	0.341E+06	0.141E+10	0.692E+09
3	0.165E+03	0.136E+03	0.297E+02	0.852E+03	0.445E+05	0.310E+06	0.124E+10	0.668E+09
3	0.333E+03	0.714E+03	0.427E+02	0.191E+04	0.114E+06	0.596E+06	0.580E+09	0.740E+09
3	0.353E+03	0.427E+03	0.520E+02	0.424E+04	0.345E+07	0.764E+07	0.147E+10	0.541E+10
3	0.407E+03	0.487E+03	0.398E+02	0.218E+04	0.481E+05	0.432E+06	0.219E+09	0.236E+10
3	0.502E+03	0.782E+03	0.486E+02	0.393E+04	0.216E+06	0.143E+07	0.232E+09	0.694E+09
3	0.441E+03	0.702E+03	0.542E+02	0.311E+04	0.268E+06	0.175E+07	0.261E+09	0.769E+09
3	0.391E+03	0.448E+03	0.299E+02	0.201E+04	0.492E+05	0.330E+06	0.187E+09	0.191E+10
3	0.390E+03	0.441E+03	0.462E+02	0.504E+04	0.279E+07	0.554E+07	0.121E+10	0.426E+10
3	0.339E+03	0.778E+03	0.490E+02	0.221E+04	0.139E+06	0.807E+06	0.700E+09	0.903E+09
3	0.113E+03	0.146E+03	0.334E+02	0.132E+03	0.201E+05	0.204E+05	0.204E+09	0.224E+09
3	0.147E+03	0.138E+03	0.414E+02	0.149E+03	0.295E+05	0.209E+05	0.294E+09	0.273E+09
3	0.150E+03	0.238E+03	0.420E+02	0.139E+03	0.512E+05	0.185E+05	0.350E+09	0.295E+09
3	0.205E+03	0.289E+03	0.337E+02	0.174E+03	0.308E+05	0.280E+05	0.465E+09	0.556E+09
3	0.212E+03	0.295E+03	0.411E+02	0.169E+03	0.436E+05	0.365E+05	0.617E+09	0.648E+09
3	0.153E+03	0.239E+03	0.294E+02	0.902E+02	0.324E+05	0.121E+05	0.250E+09	0.210E+09
3	0.138E+03	0.157E+03	0.378E+02	0.800E+02	0.222E+05	0.245E+05	0.222E+09	0.195E+09
3	0.111E+03	0.153E+03	0.303E+02	0.140E+03	0.167E+05	0.180E+05	0.184E+09	0.205E+09
3	0.235E+03	0.822E+03	0.381E+02	0.110E+04	0.255E+04	0.486E+05	0.286E+08	0.457E+08
3	0.217E+03	0.551E+03	0.320E+02	0.142E+04	0.456E+06	0.674E+06	0.124E+09	0.612E+09
3	0.256E+03	0.413E+03	0.347E+02	0.875E+03	0.156E+04	0.110E+05	0.919E+07	0.182E+09
3	0.224E+03	0.442E+03	0.279E+02	0.900E+03	0.472E+04	0.505E+05	0.871E+07	0.188E+08
3	0.246E+03	0.457E+03	0.349E+02	0.101E+04	0.614E+04	0.590E+05	0.105E+08	0.237E+08
3	0.234E+03	0.409E+03	0.292E+02	0.685E+03	0.121E+04	0.143E+05	0.850E+07	0.213E+09
3	0.240E+03	0.607E+03	0.376E+02	0.209E+04	0.425E+06	0.587E+06	0.114E+09	0.558E+09
3	0.208E+03	0.734E+03	0.289E+02	0.800E+03	0.293E+04	0.578E+05	0.336E+08	0.534E+08
3	0.141E+03	0.216E+03	0.146E+02	0.791E+02	0.181E+03	0.114E+04	0.113E+08	0.941E+07
3	0.145E+03	0.241E+03	0.197E+02	0.899E+02	0.293E+03	0.118E+04	0.145E+08	0.124E+08
3	0.148E+03	0.338E+03	0.304E+02	0.927E+02	0.209E+04	0.277E+04	0.296E+08	0.223E+08
3	0.180E+03	0.483E+03	0.418E+02	0.230E+03	0.111E+04	0.909E+04	0.138E+08	0.117E+09
3	0.176E+03	0.450E+03	0.389E+02	0.212E+03	0.106E+04	0.995E+04	0.133E+08	0.113E+09
3	0.151E+03	0.326E+03	0.279E+02	0.678E+02	0.138E+04	0.220E+04	0.238E+08	0.172E+08
3	0.137E+03	0.224E+03	0.153E+02	0.737E+02	0.274E+03	0.750E+03	0.105E+08	0.935E+07
3	0.137E+03	0.209E+03	0.151E+02	0.759E+02	0.168E+03	0.119E+04	0.103E+08	0.893E+07
3	0.175E+03	0.408E+03	0.257E+02	0.113E+03	0.462E+03	0.303E+04	0.442E+05	0.280E+06
3	0.144E+03	0.278E+03	0.276E+02	0.107E+03	0.113E+04	0.997E+03	0.982E+05	0.500E+06
3	0.157E+03	0.351E+03	0.223E+02	0.941E+02	0.527E+03	0.517E+04	0.103E+06	0.585E+06
3	0.131E+03	0.231E+03	0.274E+02	0.617E+02	0.348E+03	0.646E+03	0.463E+05	0.125E+06
3	0.128E+03	0.229E+03	0.244E+02	0.600E+02	0.272E+03	0.832E+03	0.459E+05	0.122E+06
3	0.154E+03	0.342E+03	0.216E+02	0.731E+02	0.541E+03	0.457E+04	0.113E+06	0.563E+06
3	0.148E+03	0.299E+03	0.296E+02	0.105E+03	0.112E+04	0.797E+03	0.820E+05	0.419E+06
3	0.176E+03	0.394E+03	0.248E+02	0.959E+02	0.591E+03	0.282E+04	0.488E+05	0.243E+06
3	0.156E+03	0.157E+03	0.245E+02	0.105E+03	0.208E+04	0.289E+04	0.323E+08	0.282E+08
3	0.134E+03	0.263E+03	0.295E+02	0.720E+02	0.181E+04	0.358E+04	0.264E+08	0.329E+08
3	0.122E+03	0.206E+03	0.242E+02	0.658E+02	0.353E+04	0.244E+04	0.359E+08	0.371E+08
3	0.136E+03	0.280E+03	0.321E+02	0.190E+03	0.169E+05	0.595E+04	0.680E+08	0.334E+08
3	0.137E+03	0.290E+03	0.307E+02	0.191E+03	0.156E+05	0.581E+04	0.681E+08	0.342E+08
3	0.121E+03	0.203E+03	0.263E+02	0.612E+02	0.364E+04	0.252E+04	0.313E+08	0.337E+08
3	0.136E+03	0.254E+03	0.236E+02	0.472E+02	0.225E+04	0.338E+04	0.311E+08	0.389E+08
3	0.159E+03	0.164E+03	0.224E+02	0.104E+03	0.142E+04	0.247E+04	0.253E+08	0.227E+08
3	0.163E+03	0.471E+03	0.330E+02	0.349E+03	0.324E+04	0.427E+04	0.325E+08	0.905E+08
3	0.246E+03	0.246E+03	0.327E+02	0.206E+03	0.133E+04	0.120E+05	0.368E+08	0.403E+08
3	0.217E+03	0.331E+03	0.364E+02	0.281E+03	0.274E+04	0.399E+04	0.385E+08	0.426E+08
3	0.243E+03	0.471E+03	0.435E+02	0.415E+03	0.361E+04	0.322E+05	0.124E+09	0.138E+09
3	0.251E+03	0.490E+03	0.437E+02	0.550E+03	0.147E+04	0.280E+05	0.326E+08	0.222E+09
3	0.209E+03	0.355E+03	0.348E+02	0.228E+03	0.349E+04	0.112E+05	0.407E+08	0.456E+08
3	0.231E+03	0.232E+03	0.266E+02	0.190E+03	0.121E+04	0.810E+04	0.348E+08	0.345E+08
3	0.166E+03	0.477E+03	0.314E+02	0.419E+03	0.397E+04	0.584E+04	0.536E+08	0.777E+08

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 4

CLASSIF.: DISTANCES TO CLASSES:

	CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
4	0.877E+02	0.236E+04	0.472E+07	0.230E+02	0.479E+08	0.238E+09	0.467E+12	0.239E+12
4	0.100E+03	0.200E+04	0.408E+07	0.229E+02	0.436E+08	0.216E+09	0.424E+12	0.223E+12
4	0.296E+03	0.271E+04	0.755E+07	0.311E+02	0.975E+08	0.684E+08	0.359E+12	0.185E+12
4	0.127E+03	0.148E+04	0.248E+07	0.306E+02	0.254E+08	0.950E+08	0.114E+12	0.124E+12
4	0.124E+03	0.140E+04	0.236E+07	0.312E+02	0.241E+08	0.936E+08	0.112E+12	0.120E+12
4	0.298E+03	0.237E+04	0.685E+07	0.345E+02	0.920E+08	0.624E+08	0.338E+12	0.173E+12
4	0.870E+02	0.203E+04	0.412E+07	0.274E+02	0.423E+08	0.211E+09	0.418E+12	0.213E+12
4	0.770E+02	0.208E+04	0.422E+07	0.377E+02	0.446E+08	0.220E+09	0.415E+12	0.226E+12
4	0.249E+03	0.541E+04	0.220E+09	0.440E+02	0.167E+10	0.144E+11	0.216E+13	0.258E+13
4	0.222E+03	0.973E+05	0.110E+09	0.411E+02	0.166E+11	0.848E+11	0.549E+13	0.320E+14
4	0.441E+03	0.919E+04	0.137E+09	0.523E+02	0.214E+10	0.265E+11	0.195E+13	0.218E+14
4	0.889E+03	0.439E+04	0.125E+09	0.408E+02	0.149E+10	0.197E+11	0.775E+12	0.141E+13
4	0.838E+03	0.387E+04	0.123E+09	0.324E+02	0.132E+10	0.182E+11	0.756E+12	0.124E+13
4	0.432E+03	0.853E+04	0.134E+09	0.508E+02	0.207E+10	0.257E+11	0.187E+13	0.212E+14
4	0.209E+03	0.949E+05	0.108E+09	0.404E+02	0.162E+11	0.826E+11	0.533E+13	0.312E+14
4	0.279E+03	0.399E+04	0.212E+09	0.462E+02	0.171E+10	0.139E+11	0.225E+13	0.268E+13
4	0.129E+03	0.404E+04	0.165E+08	0.270E+02	0.235E+09	0.276E+09	0.429E+12	0.406E+12
4	0.129E+03	0.444E+04	0.102E+08	0.430E+02	0.112E+09	0.218E+09	0.330E+12	0.374E+12
4	0.119E+03	0.318E+04	0.154E+08	0.310E+02	0.238E+09	0.158E+09	0.391E+12	0.384E+12
4	0.235E+03	0.142E+05	0.341E+08	0.350E+02	0.300E+09	0.249E+09	0.121E+13	0.627E+12
4	0.238E+03	0.161E+05	0.377E+08	0.350E+02	0.325E+09	0.275E+09	0.131E+13	0.684E+12
4	0.115E+03	0.240E+04	0.107E+08	0.312E+02	0.166E+09	0.109E+09	0.275E+12	0.270E+12
4	0.142E+03	0.475E+04	0.114E+08	0.347E+02	0.122E+09	0.282E+09	0.388E+12	0.436E+12
4	0.144E+03	0.339E+04	0.141E+08	0.352E+02	0.203E+09	0.266E+09	0.399E+12	0.367E+12
4	0.391E+03	0.309E+05	0.460E+09	0.262E+02	0.571E+10	0.724E+11	0.486E+13	0.564E+14
4	0.839E+03	0.226E+05	0.406E+09	0.579E+02	0.272E+10	0.394E+11	0.291E+13	0.288E+13
4	0.308E+03	0.113E+04	0.646E+09	0.481E+02	0.254E+10	0.446E+11	0.247E+13	0.346E+13
4	0.433E+03	0.353E+05	0.530E+09	0.279E+02	0.660E+10	0.836E+11	0.561E+13	0.652E+14
4	0.436E+03	0.278E+05	0.446E+09	0.377E+02	0.574E+10	0.723E+11	0.482E+13	0.564E+14
4	0.306E+03	0.111E+06	0.641E+09	0.450E+02	0.260E+10	0.439E+11	0.258E+13	0.354E+13
4	0.107E+04	0.213E+05	0.485E+09	0.577E+02	0.328E+10	0.478E+11	0.358E+13	0.354E+13
4	0.440E+03	0.307E+05	0.493E+09	0.274E+02	0.627E+10	0.792E+11	0.530E+13	0.619E+14
4	0.155E+03	0.208E+03	0.785E+02	0.155E+02	0.600E+03	0.427E+03	0.780E+07	0.736E+07
4	0.152E+03	0.216E+03	0.795E+02	0.153E+02	0.606E+03	0.389E+03	0.649E+07	0.704E+07
4	0.132E+03	0.205E+03	0.559E+02	0.223E+02	0.501E+03	0.482E+03	0.197E+08	0.113E+08
4	0.150E+03	0.341E+03	0.109E+03	0.291E+02	0.112E+04	0.963E+04	0.105E+08	0.698E+08
4	0.151E+03	0.339E+03	0.115E+03	0.306E+02	0.122E+04	0.133E+05	0.144E+08	0.767E+08
4	0.131E+03	0.204E+03	0.470E+02	0.303E+02	0.444E+03	0.103E+04	0.278E+08	0.174E+08
4	0.153E+03	0.206E+03	0.608E+02	0.150E+02	0.575E+03	0.362E+03	0.738E+07	0.806E+07
4	0.157E+03	0.225E+03	0.781E+02	0.154E+02	0.718E+03	0.340E+03	0.835E+07	0.872E+07
4	0.852E+02	0.553E+03	0.240E+07	0.345E+02	0.380E+08	0.553E+08	0.883E+11	0.777E+11
4	0.175E+03	0.436E+03	0.184E+07	0.281E+02	0.235E+08	0.306E+08	0.736E+11	0.835E+11
4	0.115E+03	0.460E+03	0.249E+07	0.316E+02	0.437E+08	0.223E+08	0.773E+11	0.750E+11
4	0.165E+03	0.106E+04	0.623E+07	0.296E+02	0.594E+08	0.398E+08	0.263E+12	0.143E+12
4	0.149E+03	0.938E+03	0.596E+07	0.338E+02	0.577E+08	0.371E+08	0.254E+12	0.138E+12
4	0.122E+03	0.452E+03	0.233E+07	0.297E+02	0.412E+08	0.202E+08	0.707E+11	0.686E+11
4	0.174E+03	0.407E+03	0.174E+07	0.302E+02	0.225E+08	0.287E+08	0.709E+11	0.807E+11
4	0.828E+02	0.434E+03	0.184E+07	0.272E+02	0.294E+08	0.390E+08	0.683E+11	0.603E+11
4	0.140E+03	0.769E+04	0.303E+08	0.355E+02	0.431E+09	0.531E+09	0.767E+12	0.727E+12
4	0.196E+03	0.105E+05	0.229E+08	0.176E+02	0.219E+09	0.747E+09	0.738E+12	0.800E+12
4	0.119E+03	0.656E+04	0.247E+08	0.253E+02	0.354E+09	0.326E+09	0.567E+12	0.557E+12
4	0.262E+03	0.263E+05	0.600E+08	0.340E+02	0.564E+09	0.482E+09	0.211E+13	0.111E+13
4	0.267E+03	0.298E+05	0.625E+08	0.406E+02	0.503E+09	0.534E+09	0.217E+13	0.116E+13
4	0.124E+03	0.629E+04	0.266E+08	0.315E+02	0.385E+09	0.344E+09	0.638E+12	0.623E+12
4	0.205E+03	0.774E+04	0.193E+08	0.321E+02	0.194E+09	0.538E+09	0.630E+12	0.698E+12
4	0.136E+03	0.544E+04	0.217E+08	0.253E+02	0.311E+09	0.360E+09	0.557E+12	0.529E+12
4	0.172E+03	0.118E+05	0.256E+08	0.274E+02	0.246E+09	0.883E+09	0.847E+12	0.916E+12
4	0.153E+03	0.134E+05	0.488E+08	0.252E+02	0.686E+09	0.739E+09	0.113E+13	0.109E+13
4	0.228E+03	0.124E+05	0.290E+08	0.250E+02	0.280E+09	0.923E+09	0.941E+12	0.102E+13
4	0.185E+03	0.286E+05	0.438E+08	0.394E+02	0.355E+09	0.162E+10	0.290E+13	0.152E+13
4	0.194E+03	0.343E+05	0.542E+08	0.425E+02	0.446E+09	0.204E+10	0.362E+13	0.191E+13
4	0.224E+03	0.129E+05	0.287E+08	0.297E+02	0.275E+09	0.920E+09	0.921E+12	0.100E+13
4	0.143E+03	0.134E+05	0.510E+08	0.250E+02	0.722E+09	0.729E+09	0.117E+13	0.114E+13
4	0.165E+03	0.142E+05	0.293E+08	0.340E+02	0.275E+09	0.105E+10	0.963E+12	0.103E+13

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 5

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
5	0.122E+03	0.191E+03	0.508E+03	0.129E+03	0.164E+02	0.108E+04	0.439E+07	0.237E+07
5	0.122E+03	0.182E+03	0.499E+03	0.140E+03	0.221E+02	0.120E+04	0.431E+07	0.256E+07
5	0.162E+03	0.340E+03	0.800E+03	0.132E+03	0.418E+02	0.378E+04	0.442E+07	0.949E+06
5	0.151E+03	0.320E+03	0.442E+03	0.705E+02	0.280E+02	0.580E+04	0.154E+07	0.315E+07
5	0.147E+03	0.300E+03	0.553E+03	0.102E+03	0.271E+02	0.519E+04	0.215E+07	0.355E+07
5	0.157E+03	0.325E+03	0.827E+03	0.130E+03	0.370E+02	0.363E+04	0.343E+07	0.588E+06
5	0.119E+03	0.173E+03	0.537E+03	0.146E+03	0.242E+02	0.114E+04	0.714E+07	0.408E+07
5	0.120E+03	0.172E+03	0.491E+03	0.137E+03	0.189E+02	0.111E+04	0.672E+07	0.316E+07
5	0.229E+03	0.387E+03	0.408E+03	0.117E+04	0.438E+02	0.142E+05	0.114E+08	0.604E+08
5	0.219E+03	0.253E+03	0.416E+03	0.104E+04	0.461E+02	0.977E+04	0.218E+08	0.238E+08
5	0.258E+03	0.535E+03	0.577E+03	0.906E+03	0.412E+02	0.740E+04	0.316E+08	0.376E+08
5	0.348E+03	0.682E+03	0.789E+03	0.217E+04	0.414E+02	0.177E+05	0.386E+08	0.155E+09
5	0.340E+03	0.675E+03	0.733E+03	0.219E+04	0.431E+02	0.159E+05	0.278E+08	0.106E+09
5	0.268E+03	0.550E+03	0.614E+03	0.103E+04	0.420E+02	0.797E+04	0.311E+08	0.377E+08
5	0.228E+03	0.252E+03	0.479E+03	0.110E+04	0.486E+02	0.103E+05	0.297E+08	0.317E+08
5	0.218E+03	0.384E+03	0.376E+03	0.114E+04	0.434E+02	0.122E+05	0.691E+07	0.343E+08
5	0.118E+03	0.234E+03	0.442E+03	0.343E+03	0.268E+02	0.356E+03	0.460E+07	0.398E+07
5	0.165E+03	0.231E+03	0.258E+03	0.235E+03	0.358E+02	0.390E+03	0.596E+07	0.528E+07
5	0.168E+03	0.312E+03	0.364E+03	0.190E+03	0.267E+02	0.506E+03	0.571E+07	0.447E+07
5	0.209E+03	0.511E+03	0.810E+03	0.190E+03	0.359E+02	0.542E+03	0.929E+07	0.800E+07
5	0.218E+03	0.519E+03	0.108E+04	0.219E+03	0.325E+02	0.829E+03	0.137E+08	0.110E+08
5	0.167E+03	0.311E+03	0.355E+03	0.213E+03	0.257E+02	0.510E+03	0.642E+07	0.502E+07
5	0.169E+03	0.241E+03	0.372E+03	0.175E+03	0.313E+02	0.622E+03	0.318E+07	0.285E+07
5	0.116E+03	0.227E+03	0.514E+03	0.353E+03	0.321E+02	0.543E+03	0.543E+07	0.491E+07
5	0.245E+03	0.377E+03	0.932E+03	0.628E+03	0.432E+02	0.118E+06	0.355E+08	0.857E+09
5	0.240E+03	0.361E+03	0.139E+04	0.466E+03	0.450E+02	0.164E+06	0.453E+08	0.511E+08
5	0.144E+03	0.800E+03	0.234E+04	0.591E+03	0.531E+02	0.207E+06	0.351E+08	0.647E+08
5	0.241E+03	0.839E+03	0.979E+03	0.504E+03	0.309E+02	0.140E+06	0.379E+08	0.912E+09
5	0.236E+03	0.878E+03	0.903E+03	0.616E+03	0.414E+02	0.141E+06	0.396E+08	0.982E+09
5	0.147E+03	0.855E+03	0.248E+04	0.671E+03	0.540E+02	0.214E+06	0.377E+08	0.720E+08
5	0.250E+03	0.372E+03	0.160E+04	0.529E+03	0.484E+02	0.203E+06	0.494E+08	0.560E+08
5	0.251E+03	0.860E+03	0.108E+04	0.590E+03	0.381E+02	0.138E+06	0.396E+08	0.895E+09
5	0.178E+03	0.373E+03	0.469E+03	0.204E+03	0.156E+02	0.759E+03	0.358E+06	0.468E+06
5	0.173E+03	0.387E+03	0.452E+03	0.190E+03	0.276E+02	0.630E+03	0.343E+06	0.366E+06
5	0.161E+03	0.386E+03	0.393E+03	0.222E+03	0.340E+02	0.292E+04	0.129E+07	0.119E+07
5	0.180E+03	0.631E+03	0.294E+03	0.173E+03	0.308E+02	0.645E+04	0.612E+06	0.426E+07
5	0.184E+03	0.639E+03	0.292E+03	0.180E+03	0.331E+02	0.710E+04	0.402E+06	0.334E+07
5	0.168E+03	0.394E+03	0.401E+03	0.260E+03	0.302E+02	0.245E+04	0.107E+07	0.978E+06
5	0.180E+03	0.382E+03	0.470E+03	0.227E+03	0.157E+02	0.703E+03	0.370E+06	0.459E+06
5	0.183E+03	0.358E+03	0.510E+03	0.237E+03	0.256E+02	0.743E+03	0.275E+06	0.344E+06
5	0.166E+03	0.334E+03	0.223E+03	0.269E+03	0.330E+02	0.351E+03	0.207E+08	0.163E+08
5	0.139E+03	0.208E+03	0.349E+03	0.309E+03	0.287E+02	0.474E+03	0.136E+08	0.107E+08
5	0.181E+03	0.271E+03	0.290E+03	0.377E+03	0.232E+02	0.346E+03	0.110E+08	0.107E+08
5	0.135E+03	0.228E+03	0.578E+03	0.209E+03	0.371E+02	0.923E+04	0.577E+08	0.259E+08
5	0.144E+03	0.211E+03	0.750E+03	0.214E+03	0.504E+02	0.211E+05	0.885E+08	0.463E+08
5	0.179E+03	0.277E+03	0.313E+03	0.395E+03	0.271E+02	0.257E+03	0.109E+08	0.106E+08
5	0.139E+03	0.214E+03	0.265E+03	0.249E+03	0.296E+02	0.476E+03	0.116E+08	0.919E+07
5	0.146E+03	0.334E+03	0.205E+03	0.281E+03	0.282E+02	0.366E+03	0.189E+08	0.144E+08
5	0.188E+03	0.277E+03	0.418E+03	0.333E+03	0.334E+02	0.187E+03	0.914E+07	0.873E+07
5	0.120E+03	0.192E+03	0.355E+03	0.242E+03	0.244E+02	0.282E+03	0.693E+07	0.501E+07
5	0.164E+03	0.316E+03	0.391E+03	0.252E+03	0.258E+02	0.556E+03	0.128E+08	0.945E+07
5	0.127E+03	0.201E+03	0.362E+03	0.206E+03	0.364E+02	0.329E+04	0.268E+08	0.257E+08
5	0.125E+03	0.197E+03	0.331E+03	0.261E+03	0.467E+02	0.568E+04	0.358E+08	0.348E+08
5	0.165E+03	0.322E+03	0.458E+03	0.270E+03	0.279E+02	0.801E+03	0.146E+08	0.106E+08
5	0.122E+03	0.205E+03	0.382E+03	0.249E+03	0.271E+02	0.281E+03	0.547E+07	0.409E+07
5	0.184E+03	0.285E+03	0.440E+03	0.353E+03	0.261E+02	0.113E+03	0.857E+07	0.819E+07
5	0.140E+03	0.301E+03	0.451E+03	0.496E+03	0.281E+02	0.891E+02	0.636E+07	0.533E+07
5	0.227E+03	0.304E+03	0.214E+03	0.300E+03	0.291E+02	0.117E+03	0.697E+07	0.654E+07
5	0.167E+03	0.275E+03	0.416E+03	0.400E+03	0.229E+02	0.293E+03	0.777E+07	0.610E+07
5	0.210E+03	0.420E+03	0.965E+03	0.251E+03	0.304E+02	0.277E+03	0.210E+08	0.121E+08
5	0.210E+03	0.420E+03	0.110E+04	0.260E+03	0.327E+02	0.367E+03	0.230E+08	0.127E+08
5	0.168E+03	0.278E+03	0.403E+03	0.402E+03	0.205E+02	0.284E+03	0.694E+07	0.544E+07
5	0.226E+03	0.310E+03	0.236E+03	0.425E+03	0.319E+02	0.127E+03	0.879E+07	0.839E+07
5	0.142E+03	0.303E+03	0.453E+03	0.533E+03	0.337E+02	0.834E+02	0.745E+07	0.610E+07

--NUMBER OF CORRECT CLASSIF. : 54
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 8

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
6	0.134E+03	0.234E+03	0.175E+03	0.2[illegible]E+03	0.396E+03	0.471E+02	0.615E+07	0.338E+07
6	0.136E+03	0.239E+03	0.172E+03	0.274E+03	0.484E+03	0.420E+02	0.482E+07	0.195E+07
6	0.184E+03	0.442E+03	0.206E+03	0.177E+03	0.477E+03	0.382E+02	0.228E+07	0.908E+06
6	0.180E+03	0.386E+03	0.152E+03	0.191E+03	0.120E+03	0.304E+02	0.125E+07	0.946E+06
6	0.180E+03	0.386E+03	0.130E+03	0.127E+03	0.948E+02	0.326E+02	0.753E+06	0.594E+06
6	0.182E+03	0.438E+03	0.282E+03	0.1[illegible]E+03	0.550E+03	0.383E+02	0.223E+07	0.992E+06
6	0.137E+03	0.246E+03	0.179E+03	0.2[illegible]E+03	0.460E+03	0.224E+02	0.438E+07	0.193E+07
6	0.135E+03	0.237E+03	0.178E+03	0.291E+03	0.371E+03	0.338E+02	0.708E+07	0.330E+07
6	0.274E+03	0.578E+03	0.302E+03	0.791E+03	0.162E+04	0.328E+02	0.276E+07	0.416E+07
6	0.212E+03	0.668E+03	0.843E+03	0.11[illegible]E+04	0.869E+04	0.344E+02	0.570E+07	0.212E+08
6	0.204E+03	0.723E+03	0.434E+03	0.125E+04	0.227E+04	0.340E+02	0.282E+06	0.746E+07
6	0.310E+03	0.593E+03	0.656E+03	0.209E+04	0.571E+04	0.577E+02	0.141E+07	0.433E+07
6	0.288E+03	0.555E+03	0.614E+03	0.191E+04	0.459E+04	0.543E+02	0.772E+06	0.241E+07
6	0.198E+03	0.702E+03	0.413E+03	0.115E+04	0.206E+04	0.303E+02	0.320E+06	0.834E+07
6	0.218E+03	0.685E+03	0.839E+03	0.12[illegible]E+04	0.101E+05	0.378E+02	0.630E+07	0.240E+08
6	0.277E+03	0.603E+03	0.325E+03	0.8[illegible]E+03	0.156E+04	0.364E+02	0.315E+07	0.491E+07
6	0.203E+03	0.746E+03	0.177E+03	0.6[illegible]E+03	0.136E+04	0.252E+02	0.243E+06	0.103E+08
6	0.218E+03	0.319E+03	0.494E+03	0.8[illegible]E+03	0.284E+04	0.343E+02	0.806E+06	0.104E+07
6	0.200E+03	0.657E+03	0.695E+03	0.3[illegible]E+03	0.746E+03	0.378E+02	0.906E+06	0.229E+07
6	0.207E+03	0.754E+03	0.199E+03	0.6[illegible]E+03	0.127E+04	0.260E+02	0.304E+06	0.115E+08
6	0.211E+03	0.825E+03	0.186E+03	0.740E+03	0.156E+04	0.57[illegible]E+02	0.689E+06	0.259E+08
6	0.197E+03	0.650E+03	0.707E+03	0.371E+03	0.731E+03	0.360E+02	0.786E+06	0.202E+07
6	0.219E+03	0.324E+03	0.494E+03	0.8[illegible]E+03	0.286E+04	0.305E+02	0.705E+06	0.922E+06
6	0.201E+03	0.718E+03	0.181E+03	0.5[illegible]E+03	0.116E+04	0.214E+02	0.301E+06	0.111E+08
6	0.143E+03	0.301E+03	0.417E+03	0.3[illegible]E+03	0.536E+02	0.317E+02	0.200E+07	0.146E+07
6	0.214E+03	0.317E+03	0.120E+03	0.2[illegible]E+03	0.206E+03	0.328E+02	0.971E+06	0.869E+06
6	0.168E+03	0.286E+03	0.334E+03	0.2[illegible]E+03	0.667E+02	0.265E+02	0.240E+07	0.177E+07
6	0.187E+03	0.455E+03	0.556E+03	0.1[illegible]E+03	0.280E+03	0.259E+02	0.393E+07	0.212E+07
6	0.184E+03	0.457E+03	0.610E+03	0.15[illegible]E+03	0.359E+03	0.340E+02	0.312E+07	0.161E+07
6	0.172E+03	0.289E+03	0.300E+03	0.2[illegible]E+03	0.875E+02	0.304E+02	0.141E+07	0.103E+07
6	0.208E+03	0.319E+03	0.116E+03	0.1[illegible]E+03	0.155E+03	0.349E+02	0.976E+06	0.881E+06
6	0.143E+03	0.304E+03	0.405E+03	0.3[illegible]E+03	0.718E+02	0.340E+02	0.130E+07	0.925E+06
6	0.202E+03	0.411E+03	0.620E+03	0.8[illegible]E+03	0.126E+03	0.133E+02	0.113E+03	0.253E+03
6	0.199E+03	0.407E+03	0.553E+03	0.7[illegible]E+03	0.122E+03	0.172E+02	0.125E+03	0.304E+03
6	0.199E+03	0.411E+03	0.539E+03	0.7[illegible]E+03	0.214E+03	0.306E+02	0.153E+03	0.419E+03
6	0.187E+03	0.455E+03	0.577E+03	0.83[illegible]E+03	0.285E+03	0.330E+02	0.499E+03	0.417E+03
6	0.185E+03	0.449E+03	0.546E+03	0.8[illegible]E+03	0.279E+03	0.313E+02	0.542E+03	0.449E+03
6	0.200E+03	0.409E+03	0.539E+03	0.8[illegible]E+03	0.219E+03	0.293E+02	0.155E+03	0.368E+03
6	0.201E+03	0.402E+03	0.561E+03	0.7[illegible]E+03	0.141E+03	0.181E+02	0.116E+03	0.298E+03
6	0.201E+03	0.413E+03	0.592E+03	0.8[illegible]E+03	0.120E+03	0.154E+02	0.129E+03	0.303E+03
6	0.145E+03	0.292E+03	0.444E+03	0.5[illegible]E+03	0.434E+02	0.332E+02	0.211E+07	0.153E+07
6	0.220E+03	0.359E+03	0.242E+03	0.5[illegible]E+03	0.120E+03	0.321E+02	0.257E+07	0.238E+07
6	0.164E+03	0.289E+03	0.422E+03	0.5[illegible]E+03	0.456E+02	0.302E+02	0.250E+07	0.184E+07
6	0.221E+03	0.482E+03	0.634E+03	0.3[illegible]E+03	0.158E+03	0.355E+02	0.540E+07	0.264E+07
6	0.225E+03	0.476E+03	0.649E+03	0.3[illegible]E+03	0.879E+02	0.262E+02	0.671E+07	0.302E+07
6	0.163E+03	0.288E+03	0.440E+03	0.6[illegible]E+03	0.344E+02	0.272E+02	0.330E+07	0.247E+07
6	0.218E+03	0.358E+03	0.250E+03	0.51[illegible]E+03	0.985E+02	0.248E+02	0.207E+07	0.190E+07
6	0.147E+03	0.297E+03	0.456E+03	0.6[illegible]E+03	0.368E+02	0.262E+02	0.215E+07	0.160E+07
6	0.136E+03	0.251E+03	0.457E+03	0.5[illegible]E+03	0.233E+03	0.229E+02	0.122E+08	0.939E+07
6	0.214E+03	0.362E+03	0.354E+03	0.5[illegible]E+03	0.606E+03	0.344E+02	0.153E+08	0.151E+08
6	0.136E+03	0.232E+03	0.422E+03	0.61[illegible]E+03	0.491E+03	0.373E+02	0.172E+08	0.138E+08
6	0.255E+03	0.433E+03	0.680E+03	0.3[illegible]E+03	0.135E+04	0.312E+02	0.442E+08	0.239E+08
6	0.258E+03	0.441E+03	0.648E+03	0.3[illegible]E+03	0.134E+04	0.325E+02	0.434E+08	0.241E+08
6	0.136E+03	0.228E+03	0.410E+03	0.5[illegible]E+03	0.300E+03	0.260E+02	0.157E+08	0.124E+08
6	0.212E+03	0.363E+03	0.341E+03	0.5[illegible]E+03	0.489E+03	0.369E+02	0.127E+08	0.123E+08
6	0.137E+03	0.264E+03	0.488E+03	0.6[illegible]E+03	0.452E+03	0.464E+02	0.157E+08	0.126E+08
6	0.142E+03	0.243E+03	0.413E+03	0.5[illegible]E+03	0.103E+03	0.277E+02	0.996E+07	0.761E+07
6	0.206E+03	0.413E+03	0.261E+03	0.5[illegible]E+03	0.354E+03	0.377E+02	0.137E+08	0.112E+08
6	0.124E+03	0.222E+03	0.435E+03	0.5[illegible]E+03	0.703E+02	0.305E+02	0.869E+07	0.628E+07
6	0.248E+03	0.482E+03	0.102E+04	0.41[illegible]E+03	0.131E+04	0.472E+02	0.193E+08	0.873E+07
6	0.240E+03	0.476E+03	0.115E+04	0.41[illegible]E+03	0.263E+04	0.528E+02	0.336E+08	0.152E+08
6	0.125E+03	0.226E+03	0.442E+03	0.61[illegible]E+03	0.753E+02	0.279E+02	0.966E+07	0.703E+07
6	0.207E+03	0.409E+03	0.293E+03	0.5[illegible]E+03	0.345E+03	0.457E+02	0.180E+08	0.148E+08
6	0.142E+03	0.243E+03	0.446E+03	0.6[illegible]E+03	0.149E+03	0.316E+02	0.144E+08	0.111E+08

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 7

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
7	0.156E+03	0.348E+03	0.244E+03	0.500E+03	0.315E+03	0.261E+03	0.533E+02	0.818E+02
7	0.157E+03	0.353E+03	0.240E+03	0.550E+03	0.299E+03	0.237E+03	0.317E+02	0.209E+03
7	0.180E+03	0.412E+03	0.317E+03	0.533E+03	0.312E+03	0.330E+03	0.389E+02	0.710E+02
7	0.199E+03	0.401E+03	0.227E+03	0.527E+03	0.129E+03	0.185E+03	0.204E+02	0.743E+02
7	0.191E+03	0.401E+03	0.223E+03	0.516E+03	0.129E+03	0.206E+03	0.314E+02	0.690E+02
7	0.181E+03	0.419E+03	0.314E+03	0.516E+03	0.324E+03	0.348E+03	0.285E+02	0.683E+02
7	0.158E+03	0.358E+03	0.254E+03	0.502E+03	0.324E+03	0.270E+03	0.192E+02	0.788E+02
7	0.159E+03	0.359E+03	0.252E+03	0.509E+03	0.299E+03	0.236E+03	0.233E+02	0.963E+02
7	0.227E+03	0.456E+03	0.442E+03	0.737E+03	0.137E+04	0.480E+03	0.322E+02	0.119E+03
7	0.182E+03	0.551E+03	0.760E+03	0.740E+03	0.113E+04	0.841E+03	0.526E+02	0.243E+03
7	0.174E+03	0.549E+03	0.588E+03	0.708E+03	0.674E+03	0.482E+03	0.337E+02	0.410E+03
7	0.222E+03	0.406E+03	0.498E+03	0.909E+03	0.108E+04	0.586E+03	0.472E+02	0.368E+03
7	0.221E+03	0.423E+03	0.478E+03	0.935E+03	0.113E+04	0.650E+03	0.470E+02	0.104E+03
7	0.172E+03	0.526E+03	0.605E+03	0.717E+03	0.677E+03	0.382E+03	0.366E+02	0.333E+03
7	0.181E+03	0.534E+03	0.799E+03	0.717E+03	0.104E+04	0.741E+03	0.545E+02	0.815E+03
7	0.224E+03	0.458E+03	0.472E+03	0.754E+03	0.125E+04	0.459E+03	0.348E+02	0.234E+03
7	0.175E+03	0.372E+03	0.316E+03	0.600E+03	0.277E+03	0.318E+03	0.353E+02	0.339E+03
7	0.203E+03	0.387E+03	0.403E+03	0.681E+03	0.202E+03	0.523E+03	0.515E+02	0.247E+03
7	0.189E+03	0.361E+03	0.279E+03	0.644E+03	0.542E+03	0.245E+03	0.326E+02	0.866E+02
7	0.194E+03	0.464E+03	0.463E+03	0.618E+03	0.362E+03	0.342E+03	0.482E+02	0.199E+03
7	0.190E+03	0.456E+03	0.463E+03	0.642E+03	0.355E+03	0.330E+03	0.322E+02	0.147E+03
7	0.191E+03	0.364E+03	0.287E+03	0.657E+03	0.511E+03	0.213E+03	0.271E+02	0.896E+02
7	0.202E+03	0.383E+03	0.363E+03	0.661E+03	0.180E+03	0.418E+03	0.367E+02	0.139E+03
7	0.177E+03	0.375E+03	0.350E+03	0.698E+03	0.273E+03	0.268E+03	0.357E+02	0.291E+03
7	0.194E+03	0.423E+03	0.673E+03	0.933E+03	0.207E+03	0.966E+02	0.134E+02	0.825E+02
7	0.195E+03	0.422E+03	0.656E+03	0.933E+03	0.202E+03	0.862E+02	0.120E+02	0.723E+02
7	0.194E+03	0.419E+03	0.639E+03	0.935E+03	0.194E+03	0.826E+02	0.167E+02	0.891E+02
7	0.197E+03	0.417E+03	0.608E+03	0.942E+03	0.166E+03	0.399E+02	0.307E+02	0.390E+02
7	0.197E+03	0.421E+03	0.615E+03	0.951E+03	0.159E+03	0.325E+02	0.283E+02	0.568E+02
7	0.194E+03	0.419E+03	0.634E+03	0.959E+03	0.202E+03	0.756E+02	0.806E+02	0.836E+02
7	0.196E+03	0.423E+03	0.647E+03	0.944E+03	0.182E+03	0.745E+02	0.972E+01	0.550E+02
7	0.195E+03	0.415E+03	0.635E+03	0.934E+03	0.194E+03	0.917E+02	0.162E+02	0.698E+02
7	0.203E+03	0.675E+03	0.464E+03	0.517E+03	0.327E+03	0.468E+03	0.371E+02	0.538E+05
7	0.217E+03	0.327E+03	0.933E+03	0.634E+03	0.131E+04	0.910E+03	0.469E+02	0.342E+04
7	0.197E+03	0.521E+03	0.562E+03	0.563E+03	0.416E+03	0.291E+04	0.552E+02	0.322E+04
7	0.172E+03	0.564E+03	0.381E+03	0.403E+03	0.420E+03	0.356E+03	0.565E+02	0.120E+05
7	0.177E+03	0.593E+03	0.446E+03	0.473E+03	0.381E+03	0.318E+03	0.587E+02	0.556E+05
7	0.195E+03	0.529E+03	0.535E+03	0.471E+03	0.393E+03	0.314E+04	0.577E+02	0.660E+04
7	0.218E+03	0.326E+03	0.897E+03	0.643E+03	0.134E+04	0.946E+03	0.488E+02	0.358E+04
7	0.204E+03	0.659E+03	0.451E+03	0.506E+03	0.296E+03	0.479E+03	0.429E+02	0.423E+05
7	0.187E+03	0.430E+03	0.597E+03	0.736E+03	0.269E+03	0.101E+03	0.405E+02	0.891E+03
7	0.214E+03	0.389E+03	0.410E+03	0.734E+03	0.401E+03	0.977E+02	0.283E+02	0.187E+03
7	0.213E+03	0.374E+03	0.475E+03	0.736E+03	0.289E+03	0.106E+03	0.308E+02	0.266E+03
7	0.175E+03	0.431E+03	0.618E+03	0.749E+03	0.510E+03	0.461E+03	0.382E+02	0.394E+04
7	0.175E+03	0.427E+03	0.623E+03	0.737E+03	0.513E+03	0.439E+03	0.291E+02	0.293E+04
7	0.214E+03	0.375E+03	0.472E+03	0.733E+03	0.308E+03	0.115E+03	0.284E+02	0.260E+03
7	0.213E+03	0.395E+03	0.450E+03	0.742E+03	0.322E+03	0.697E+02	0.243E+02	0.159E+03
7	0.194E+03	0.429E+03	0.552E+03	0.736E+03	0.282E+03	0.137E+03	0.303E+02	0.871E+03
7	0.183E+03	0.381E+03	0.392E+03	0.695E+03	0.250E+03	0.170E+03	0.263E+02	0.997E+02
7	0.208E+03	0.399E+03	0.335E+03	0.669E+03	0.180E+03	0.278E+03	0.237E+02	0.153E+03
7	0.186E+03	0.350E+03	0.323E+03	0.664E+03	0.376E+03	0.131E+03	0.199E+02	0.156E+03
7	0.185E+03	0.425E+03	0.470E+03	0.688E+03	0.319E+03	0.283E+03	0.276E+02	0.276E+03
7	0.184E+03	0.426E+03	0.451E+03	0.644E+03	0.341E+03	0.299E+03	0.308E+02	0.385E+03
7	0.188E+03	0.353E+03	0.331E+03	0.700E+03	0.386E+03	0.119E+03	0.290E+02	0.144E+03
7	0.208E+03	0.402E+03	0.348E+03	0.644E+03	0.167E+03	0.408E+03	0.255E+02	0.124E+03
7	0.183E+03	0.381E+03	0.376E+03	0.677E+03	0.259E+03	0.187E+03	0.262E+02	0.924E+02
7	0.193E+03	0.359E+03	0.412E+03	0.692E+03	0.241E+03	0.865E+02	0.226E+02	0.154E+03
7	0.191E+03	0.423E+03	0.350E+03	0.630E+03	0.113E+03	0.404E+02	0.271E+02	0.113E+03
7	0.182E+03	0.379E+03	0.434E+03	0.674E+03	0.217E+03	0.984E+02	0.226E+02	0.244E+03
7	0.185E+03	0.418E+03	0.415E+03	0.725E+03	0.338E+03	0.160E+03	0.292E+02	0.183E+03
7	0.187E+03	0.426E+03	0.420E+03	0.728E+03	0.308E+03	0.114E+03	0.441E+02	0.211E+03
7	0.186E+03	0.377E+03	0.449E+03	0.708E+03	0.214E+03	0.855E+02	0.235E+02	0.156E+03
7	0.192E+03	0.424E+03	0.375E+03	0.657E+03	0.120E+03	0.560E+02	0.297E+02	0.753E+02
7	0.193E+03	0.359E+03	0.402E+03	0.695E+03	0.256E+03	0.886E+02	0.193E+02	0.135E+03

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.00000

RESULTS FOR CLASS 8

CLASSIF.:	DISTANCES TO CLASSES: CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 5	CLASS 6	CLASS 7	CLASS 8
8	0.159E+03	0.369E+03	0.284E+03	0.611E+03	0.366E+03	0.200E+03	0.104E+03	0.258E+02
8	0.162E+03	0.376E+03	0.302E+03	0.631E+03	0.357E+03	0.177E+03	0.166E+03	0.428E+02
8	0.178E+03	0.417E+03	0.330E+03	0.550E+03	0.349E+03	0.275E+03	0.132E+03	0.313E+02
8	0.193E+03	0.397E+03	0.242E+03	0.541E+03	0.130E+03	0.197E+03	0.513E+02	0.349E+02
8	0.193E+03	0.393E+03	0.251E+03	0.531E+03	0.135E+03	0.189E+03	0.682E+02	0.291E+02
8	0.180E+03	0.415E+03	0.360E+03	0.592E+03	0.354E+03	0.300E+03	0.193E+03	0.359E+02
8	0.159E+03	0.361E+03	0.297E+03	0.632E+03	0.383E+03	0.245E+03	0.126E+03	0.303E+02
8	0.158E+03	0.363E+03	0.289E+03	0.630E+03	0.382E+03	0.214E+03	0.123E+03	0.401E+02
8	0.228E+03	0.447E+03	0.453E+03	0.804E+03	0.111E+04	0.379E+03	0.532E+02	0.454E+02
8	0.185E+03	0.548E+03	0.822E+03	0.771E+03	0.932E+03	0.684E+03	0.173E+03	0.434E+02
8	0.173E+03	0.522E+03	0.652E+03	0.754E+03	0.796E+03	0.373E+03	0.114E+03	0.382E+02
8	0.227E+03	0.412E+03	0.554E+03	0.943E+03	0.966E+03	0.466E+03	0.668E+02	0.323E+02
8	0.223E+03	0.414E+03	0.533E+03	0.927E+03	0.910E+03	0.412E+03	0.442E+02	0.393E+02
8	0.173E+03	0.525E+03	0.668E+03	0.763E+03	0.885E+03	0.416E+03	0.208E+03	0.460E+02
8	0.185E+03	0.561E+03	0.834E+03	0.776E+03	0.111E+04	0.812E+03	0.299E+03	0.431E+02
8	0.233E+03	0.450E+03	0.433E+03	0.811E+03	0.128E+04	0.439E+03	0.817E+02	0.429E+02
8	0.187E+03	0.361E+03	0.324E+03	0.652E+03	0.435E+03	0.273E+03	0.103E+03	0.316E+02
8	0.201E+03	0.413E+03	0.441E+03	0.652E+03	0.157E+03	0.287E+03	0.119E+03	0.332E+02
8	0.185E+03	0.370E+03	0.347E+03	0.660E+03	0.476E+03	0.258E+03	0.114E+03	0.275E+02
8	0.189E+03	0.464E+03	0.445E+03	0.637E+03	0.401E+03	0.211E+03	0.126E+03	0.323E+02
8	0.191E+03	0.466E+03	0.468E+03	0.636E+03	0.404E+03	0.227E+03	0.150E+03	0.278E+02
8	0.187E+03	0.372E+03	0.354E+03	0.607E+03	0.471E+03	0.192E+03	0.844E+02	0.302E+02
8	0.201E+03	0.412E+03	0.444E+03	0.648E+03	0.165E+03	0.331E+03	0.137E+03	0.242E+02
8	0.187E+03	0.366E+03	0.338E+03	0.634E+03	0.400E+03	0.265E+03	0.101E+03	0.284E+02
8	0.197E+03	0.563E+03	0.749E+03	0.788E+03	0.317E+03	0.251E+03	0.349E+03	0.475E+02
8	0.224E+03	0.382E+03	0.698E+03	0.929E+03	0.653E+03	0.343E+03	0.158E+03	0.434E+02
8	0.208E+03	0.458E+03	0.521E+03	0.834E+03	0.286E+03	0.801E+03	0.368E+03	0.566E+02
8	0.180E+03	0.513E+03	0.700E+03	0.790E+03	0.389E+03	0.191E+03	0.786E+02	0.272E+02
8	0.179E+03	0.523E+03	0.667E+03	0.759E+03	0.402E+03	0.223E+03	0.756E+02	0.444E+02
8	0.208E+03	0.455E+03	0.505E+03	0.832E+03	0.286E+03	0.755E+03	0.429E+03	0.535E+02
8	0.221E+03	0.376E+03	0.739E+03	0.926E+03	0.529E+03	0.249E+03	0.756E+02	0.543E+02
8	0.196E+03	0.584E+03	0.758E+03	0.787E+03	0.435E+03	0.354E+03	0.603E+03	0.529E+02
8	0.202E+03	0.438E+03	0.841E+03	0.110E+04	0.207E+03	0.474E+02	0.762E+02	0.132E+02
8	0.203E+03	0.438E+03	0.847E+03	0.111E+04	0.202E+03	0.430E+02	0.689E+02	0.121E+02
8	0.204E+03	0.440E+03	0.845E+03	0.111E+04	0.205E+03	0.631E+02	0.731E+02	0.234E+02
8	0.206E+03	0.453E+03	0.828E+03	0.111E+04	0.191E+03	0.531E+02	0.811E+02	0.195E+02
8	0.206E+03	0.452E+03	0.803E+03	0.109E+04	0.185E+03	0.482E+02	0.726E+02	0.196E+02
8	0.206E+03	0.441E+03	0.846E+03	0.111E+04	0.195E+03	0.618E+02	0.655E+02	0.307E+02
8	0.203E+03	0.437E+03	0.820E+03	0.109E+04	0.188E+03	0.404E+02	0.924E+02	0.147E+02
8	0.203E+03	0.435E+03	0.845E+03	0.111E+04	0.205E+03	0.445E+02	0.682E+02	0.159E+02
8	0.189E+03	0.392E+03	0.464E+03	0.813E+03	0.255E+03	0.289E+03	0.767E+02	0.339E+02
8	0.213E+03	0.411E+03	0.529E+03	0.779E+03	0.192E+03	0.410E+03	0.985E+02	0.383E+02
8	0.197E+03	0.379E+03	0.438E+03	0.804E+03	0.347E+03	0.172E+03	0.588E+02	0.202E+02
8	0.197E+03	0.464E+03	0.616E+03	0.788E+03	0.431E+03	0.359E+03	0.138E+03	0.294E+02
8	0.194E+03	0.462E+03	0.590E+03	0.745E+03	0.464E+03	0.420E+03	0.157E+03	0.298E+02
8	0.197E+03	0.384E+03	0.462E+03	0.829E+03	0.308E+03	0.158E+03	0.669E+02	0.236E+02
8	0.214E+03	0.416E+03	0.574E+03	0.827E+03	0.212E+03	0.396E+03	0.118E+03	0.198E+02
8	0.191E+03	0.394E+03	0.498E+03	0.857E+03	0.240E+03	0.268E+03	0.735E+02	0.258E+02
8	0.187E+03	0.374E+03	0.444E+03	0.770E+03	0.183E+03	0.150E+03	0.225E+03	0.381E+02
8	0.208E+03	0.412E+03	0.472E+03	0.767E+03	0.157E+03	0.181E+03	0.261E+03	0.254E+02
8	0.191E+03	0.369E+03	0.446E+03	0.779E+03	0.223E+03	0.673E+02	0.140E+03	0.283E+02
8	0.196E+03	0.445E+03	0.546E+03	0.781E+03	0.336E+03	0.223E+03	0.760E+03	0.551E+02
8	0.195E+03	0.441E+03	0.553E+03	0.810E+03	0.317E+03	0.185E+03	0.221E+03	0.369E+02
8	0.193E+03	0.373E+03	0.468E+03	0.809E+03	0.208E+03	0.556E+02	0.232E+03	0.337E+02
8	0.208E+03	0.412E+03	0.488E+03	0.779E+03	0.156E+03	0.166E+03	0.256E+03	0.339E+02
8	0.190E+03	0.379E+03	0.478E+03	0.790E+03	0.168E+03	0.987E+02	0.938E+02	0.311E+02
8	0.205E+03	0.442E+03	0.614E+03	0.820E+03	0.260E+03	0.178E+03	0.584E+03	0.315E+02
8	0.199E+03	0.374E+03	0.422E+03	0.790E+03	0.352E+03	0.148E+03	0.931E+02	0.306E+02
8	0.214E+03	0.408E+03	0.492E+03	0.800E+03	0.182E+03	0.243E+03	0.170E+03	0.243E+02
8	0.168E+03	0.414E+03	0.529E+03	0.803E+03	0.493E+03	0.123E+03	0.718E+03	0.333E+02
8	0.166E+03	0.412E+03	0.500E+03	0.820E+03	0.513E+03	0.138E+03	0.617E+03	0.321E+02
8	0.214E+03	0.410E+03	0.495E+03	0.794E+03	0.190E+03	0.288E+03	0.356E+03	0.336E+02
8	0.199E+03	0.373E+03	0.414E+03	0.795E+03	0.388E+03	0.191E+03	0.122E+03	0.266E+02
8	0.203E+03	0.439E+03	0.576E+03	0.781E+03	0.234E+03	0.162E+03	0.427E+03	0.325E+02

--NUMBER OF CORRECT CLASSIF. : 64
NUMBER OF ERRONEOUS CLASSIF. : 0
PERCENTAGE OF SUCCESS : 100.000000